ubuntu

笨兔兔的故事

作者：**懒蜗牛 Gentoo**

技术支持：
http://www.ubuntu.com/
http://forum.ubuntu.org.cn/

Lucid Lynx 10.04 LTS

Ubuntu is an African word meaning 'Humanity to others', or 'I am what I am because of who we all are'. The Ubuntu distribution brings the spirit of Ubuntu to the software world.

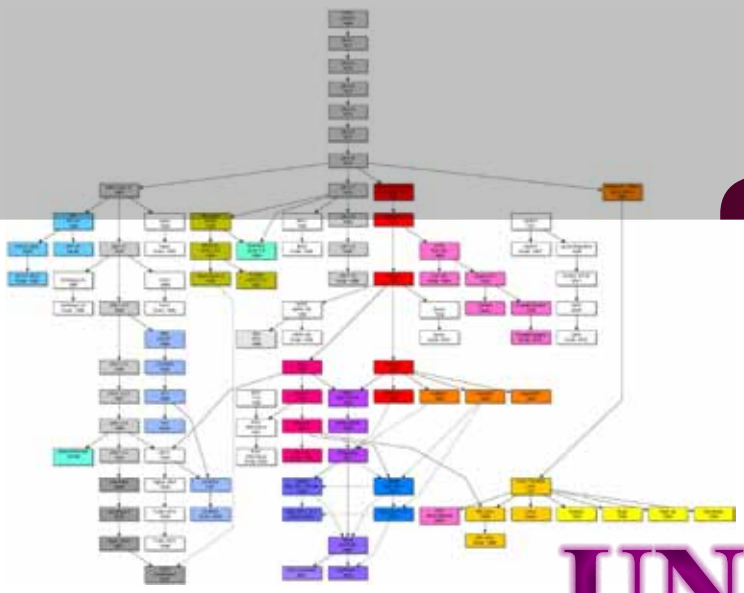
ubuntu

Lucid Lynx 10.04 LTS

我叫 ubuntu，主人喜欢叫我“笨兔”，但是我绝对不笨，与某种耳朵长尾巴短的哺乳动物也没有什么联系，我是一个操作系统，我是一个 ***Linux***，我是 ubuntu。

Chapter 0

序
—一切的起因



我们的故事从上个世纪 60 年代的美国开始。

那个时代的计算机是个新鲜玩意，非常笨重，家庭用户是没有的，都是商用或者试验，科学计算用的机器。你说你想买个电脑斗地主？把你卖给地主你也买不起呀。再说那时候的计算机不是随便一个人就会用的，那时候的计算机使用的时候是由人来输入一条条的指令，来进行各种运算的。他们输入的指令大约相当于现在的汇编指令，所以这个效率和操作难度有多高就可想而知了。那时候计算机大都没有什么操作系统，顶多有个批处理系统，可以把要输入的指令记录在某种媒介上（比如纸带）一次性输入进去，让人们省去一条条重复输入指令的麻烦。后来慢慢有了很简单的操作系统，但并不像现在我们见到的操作系统这样通用。这个时候，卖计算机的厂商要为每一型号的计算机设计不同的操作系统，一个程序如果在这个型号的计算机上写好了，拿到其他型号的计算机上是运行不了的，因为这两台机器连操作系统都不一样，怎么可能程序通用呢。

计算机要是老这样肯定是不行啦，否则你今天要玩斗地主，人家游戏公司就得专门派人到你家机器上现写出一个来——因为不同型号的计算机上的操作系统不同用嘛。这个斗地主的问题，终于还是被那个时代 IT 业界的大地主，蓝色的 IBM 公司率先着手解决了。1964 年他们公司推出了一个系列的大型机，用途、价位，各不一样，但他们上面运行的操作系统，都是 System/360。（这 360 可不是卖鞋的，也不是跟 QQ 打架的那个。）这一下获得了很大的成功，因为省去了为每一台电脑单独编写系统的成本嘛。直到今天，IBM 的大型机上依然可以运行这个 360 系统，可见其当初设计时充分考虑的兼容性。然而我们要讲的主角不是 360，而是另一个伟大的操作系统。

那时候有个聚集了很多牛人的地方，叫做贝尔实验室，是 1925 年由 AT&T 公司成立的。一帮头脑发达四肢也不一定简单的家伙整天聚在那里，研究新奇的东西，什么任意门啊，竹蜻蜓啊，记忆面包啊——呃……都不是他们发明的（发明这些的人是个日本科学家）。那贝尔实验室那帮人研究什么呢？贝尔实验室的工作可以大致分为三个类别：基础研究，系统工程和应用开发。在基础研究方面主要从事电信技术的基础理论研究，包括数学、物理学、材料科学、行为科学和计算机编程理论，反正都是大学听不懂的那几门就对了。系统工程主要研究构成电信网络的高度复杂系统。开发部门是贝尔实验室最大的部门，负责设计构成贝尔系统电信网络的设备和软件。具体来说贝尔实验室研究出来过的东西有晶体管、发光二极管、数字交换机、通信卫星、电子数字计算机、蜂窝移动通信、有声电影、立体声录音，等等。（怎么样，不比机器猫那些东西差吧？）通信网的许多重大发明都诞生自这里。

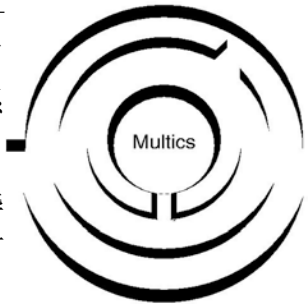


那时候还有个聚集了很多牛人的地方，叫做麻省理工学院（MIT），这是美国的一所综合性私立大学，有“世界理工大学之最”的美名。从这里走出的牛人很多，到 2009 年为止，先后有 76 位诺贝尔奖得住，都曾经在麻省理工学院学习或者工作。麻省理工学院的自然及工程科学在世界上享有极佳的盛誉，其管理学、经济学、哲学、政治学、语言学也同样优秀。另外，麻省理工研发高科技武器和美国最高机密的林肯实验室、领先世界一流的计算机科学及人工智能实验室、世界尖端的媒体实验室、和培养了许多全球顶尖首席执行官斯隆管理学院也都是麻省理工赫赫有名宝贵资产。有着毋庸置疑的实力，麻省理工自然非常不差钱，截止 2008 年底麻省理工有 101 亿美元的总资产，因为不差钱，于是该校对家庭年收入低于 75000 美元的学生一律免学费。哎，万恶的资本主义阿……不提也罢。

那时候，又有个聚集了很多牛人的地方。（哪那么多地方阿！）这地方是个公司，叫做通用电气。这公司可是个大公司，当年是个

卖灯泡的，他们的灯泡可非同一般，虽然不节能，虽然寿命不如现在的长，虽然价格比现在贵，虽然外形也不一定好看，但是——他们是第一家卖灯泡的！因为他们的老大，就是大名鼎鼎的托马斯 . 爱迪生。1876 年，发明灯泡的爱迪生同学成立了爱迪生灯泡厂，为节约蜡烛和灯油做出了突出的贡献，估计那年的五一劳动奖章肯定是他的了。到 1890 年，爱迪生同学将灯泡厂重组，成立的爱迪生通用电气公司，到 1892 年又与汤姆森－休斯顿电气公司合并，成立了通用电气公司。

好，时间到了 1965 年，这三个聚集着不少牛人的地方有一天忽然想合作一把。于是，大名鼎鼎的贝尔实验室，大名鼎鼎的麻省理工学院和大名鼎鼎的通用电气公司一起开始了一个制作操作系统的计划。为了结束长期以来计算机上面没有统一的操作系统的混乱局面，他们决定要创造出一套前无古人后无来者，上得厅堂，下得厨房，念天地之悠悠，独怆然而泣下的惊世骇俗的操作系统！具体来说吧，这个操作系统应该是一个支持多使用者、多任务、多层次的操作系统，因为这三多，所以这个操作系统就起名叫做——MULTICS。（难道你以为叫许三多？人家讲的是英语好不好。）有了这三家的强强联合，那开发的结果还用问么？这个 MULTICS 操作系统的项目在 1965 年成立，到了 1969 年就……被取消了。



咳咳，这个……其实编写操作系统也不是一件容易的事儿啦。毕竟道路是曲折滴，研究是辛苦滴，成绩还是有滴，失败捏……也是可以原谅滴嘛。

项目失败了，大家都很沮丧。在这些沮丧的人中，Kenneth Lane Thompson 只是很普通的一个。Kenneth Lane Thompson 是 1943 年出生在美国的新奥尔良的，吃着烤翅长大的他，没有辜负养育他长大的父母和那些没有了翅膀的鸡。1960 年，汤普逊考上了加州大学博克莱分校主修电气工程，顺利取得了电子工程硕士的学位。1966 年，他加入了贝尔实验室，参与了 MULTICS 项目。做项目是个很辛苦的事情，在疲劳的揉揉因熬夜而发红的眼睛后，他很想能有个电脑游戏来玩玩。然而那时候别说超级玛丽，连吃豆也没有啊！所以汤普逊同学就自己编了一个游戏，叫做星际旅行。这个星际旅行跟星际争霸那肯定是没的比的了，不过在那时候已经算是很有吸引力了。这个游戏自然是被设计运行在 MULTICS 系统上的，由于 MULTICS 系统还不完善，所以游戏运行的也不是很流畅，所以，能够顺畅的玩星际旅行，成为了汤普逊同学努力工作的动力。可是后来项目被干掉了，如果事情就这样结束，那么汤普逊同学就再也不可能流畅的玩他的星际旅行了，这是多么遗憾的事情阿。可是现实是残酷的，项目确实就是取消了，要想顺畅的玩游戏怎么办？毛主席教导我们说：自己动手，丰衣足食。我估计汤普逊没有背过毛泽东语录，但是他用自己的行动证明了这两句话的正确性。他在墙角淘换出一台 PDP-7 的机器，并且伙同其同事 Dernis Ritchie，打算将星际旅行移植到这台 PDP-7 上。

当然，要想运行这游戏，还是得有个系统。有了固定的系统，那以后再编写别的游戏就

更方便了。可是系统从哪里来？ MULTICS ？已经停工了，并且这系统绝对不是两个人可以搞定的。那怎么办？还得自己动手！于是 Ken Thompson 和 Dernis Ritchie 再次发扬自己动手的精神，用汇编语言写出来个系统，这就是最初的，非常简陋的，UNIX 的前身。这个系统不像 MULTICS 那么牛，不支持很多的用户，只能支持两个用户，（估计是为了避免做好了之后俩人抢机器玩的局面发生。也可能是为了以后俩人对战？）支持的进程也有限，其他功能也都没有 MULTICS 设计的那么复杂。相对于那个 MULTICS 系统——MULTiplexed Information and Computing System，Brian Kernighan 开玩笑地戏称他们的系统其实是："UNiplexed Information and Computing System"，缩写为 "UNICS"。后来大家取其谐音，就诞生了 UNIX 这个词。这一年，已经是 1970 年，史称 Unix 元年。直到现在，计算机中都是用 1970 年 1 月 1 日 0 点 0 分 0 秒为原点来记录时间。（计算机中的时间记录的是自 1970 年 1 月 1 日 0 点 0 分 0 秒开始，到现在经过的总秒数，再用这个秒数计算出年，月，日）后来，Brian Kernighan 觉得用汇编写的系统不好维护，于是……他发明了 C 语言（符合大牛一切自己动手的风格），然后用 C 语言又重写了一遍。从此，Unix 走上了发展的快车道，并且一直用到现在。许多世界级的大服务器，用的都是 Unix 系统。

而这一切的努力，就是为了玩个游戏。-_-b

Ken (seated) and Dennis (standing) at a PDP-11 in 1972

这回要说的，是另一个传奇人物。

Richard Stallman,1953 年出生在美国纽约曼哈顿地区，他从一出生就……没什么特别；他上小学的时候……反正我不认识他；等到他上初中的时候呢……也还没我呢。总之，他在生命的前十几年中并没有表现出什么过人的地方，因为他没遇到一个叫做电脑的东西。

高中的一个暑假，他去给 IBM 打工，花了两周的时间用 Fortran 语言编了一个数据处理的程序。这是他第一次接触计算机，或许就是这次相遇，确定了他未来行走的方向。后来，1971 年，他考上了哈佛大学，听说这学校不错，怎么也得是个区重点吧。上学的同时，他还受聘于麻省理工学院的人工智能实验室，成为了一名职业黑客（黑客这个词没有贬义）。也不知道他哪来的那么多时间，可能也是把毛概和邓论都翘了吧。在人工智能实验室的期间，他可没少干活，开发了很多有用的软件，其中最著名的就是 Emacs 编辑器。Emacs 是一个可与 vi 相抗衡的强大的编辑器，他们俩的操作方式完全不同，但却同样强大，各自用自己独有的方式，提高着人们的编辑效率。直到今天，仍然总有人争论到底 emacs 好还是 vi 好，信奉 emacs 的人和信奉 vi 的人形成了两个帮派，这俩帮派经常在大街上用板砖菜刀拼个你死我活。哦，扯远了，咱还回来说 Stallman。

那时候的 Stallman 在人工智能实验室里工作的非常 Happy，大家有 BUG 同担，有代码共享。软件工程师的世界，是一个人人为我，我为人人的世界。因为咱说过，最初的计算机就像我们的算盘一样，只是一个硬件， 没有软件的概念。后来随着电子管、晶体管的发明，计算机的电子成分才超越了机械成分，逐步演化成了现在的电子计算机，在这个过程中，出现了软件，并起到越来越重要的作用，最终成为了

计算机的灵魂。而最初的计算机软件没有什么开源不开源的概念，因为那时候软件天生就是自由的！卖计算机的同时会附带软件，包括软件的源代码和文档。用户可以根据自己的需要去进行修改软件，与别人分享软件，总之，软件是用户花钱买硬件时附带着买来的，用户想怎么玩就怎么玩。软件开发者的目的，也不是靠软件赚钱，而是靠软件支撑起硬件的功能，然后卖硬件赚钱。然而随着技术的发展，软件逐渐脱离硬件成为一个独立的产业，很多软件慢慢的只提供二进制代码而不提供源码了，这就意味着你不能修改它，并且多数还规定最终用户没有二次分发的权利。也就是说，这东西你买了，只能你用，你再给别人，不行！这就好像我买了把菜刀，然后卖菜刀的告诉我“你这把菜刀不许借给你的邻居用，也不许私给菜刀换刀把，否则我就告你！”……囧，你管的着么！？

Stallman 当时就遇到了类似这样的菜刀问题。那时候，他们实验室买的第一台打印机附带有驱动程序的源代码。他们那的黑客们可以随意修改这个驱动，根据自己的需要添加些小功能啊，改改 bug 啊之类的，这为他们的工作带来了很大的方便。后来，实验室又买了一台激光打印机，这次厂商只提供了二进制的打印机驱动程序，它是实验室里仅有的一个没有源代码的软件。出于工作的需要，Richard Stallman 想修改一下这个驱动程序，但是不行啊，没源码啊。后来 Richard Stallman 听说卡内基·梅隆大学有这个打印机的驱动程序源代码，他就去了那里，对他们说：“那啥，大家都是道上混的，谁还没个马高蹬短的时候？是兄弟的拉哥们一把，我也没啥事儿，就是我们那打印机老丢字，一遇到什么敏感的字眼就给我打成口口，我估计是驱动的问题，听说你们这有这驱动的源码，能不能给我拷一份？”对方办事效率还是挺高的，很干脆的拒绝了他。因为他们和厂商签署了一份保密协议，协议要求他们不能向别人拷贝源代码。顿时 Richard Stallman 感到他们背叛了自由的计算机社团，他非常生气，但是他选择了沉默。这只是一件小事，只是一个时代的缩影。那个时代，正处在软件向私有化转变的过程中，越来越多的软件选择了不开放源代码，不允许二次分发的发布方式。甚至 Stallman 身边的同志们也都一个一个都跑到那些靠卖私有软件挣钱的公司去打工了。而 Stallman 依然沉默。

不在沉默中爆发，就在沉默中灭亡。

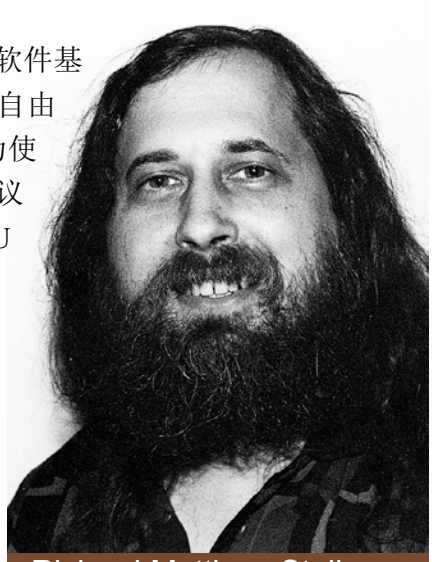
Stallman 爆发了！

他不能容忍软件世界里清新自由的空气被私有软件污染的乌烟瘴气；他不能容忍被剥夺按照自己的需求修改软件的权利和乐趣；他不能容忍自己买条皮带尺寸不够，他竟然连自己在上面多打个洞的权利都没有！

于是，他爆发了。

他要重现当年那人人为我，我为人人的合作互助的软件世界；他要把使用、复制、研究、修改、分发软件的权利还给每一个软件世界的人民；他要用自己的行动告诉人们，软件天生就该是自由的！他要开辟一个新的世界，哪怕是一个人在战斗！于是，一个宏伟的计划在他心中产生——GNU 计划。它的目标是创建一套完全自由的操作系统，因为操作系统是电脑中最重要的最基础的软件，要创造自由的软件世界，自然先要有一套自由的操作系统，然后再以此系统为中心，开发各种各样自由的软件。Richard Stallman 最早是在 net.unix-wizards 新闻

组上公布了 GNU 计划，那是 1983 年的事情。既然要做操作系统，首先得有个明确的规划和目标，目标是什么？这个操作系统要做成什么样子？这当然是要向最成功的操作系统学习，哪个？ UNIX ！ GNU 计划中的操作系统，将是一个类 Unix 的操作系统。这个系统要使用与 Unix 相同的接口标准，这样，就可以由不同的人，分期分批的创作操作系统的不同部分而不必担心相互之间协同工作的问题。

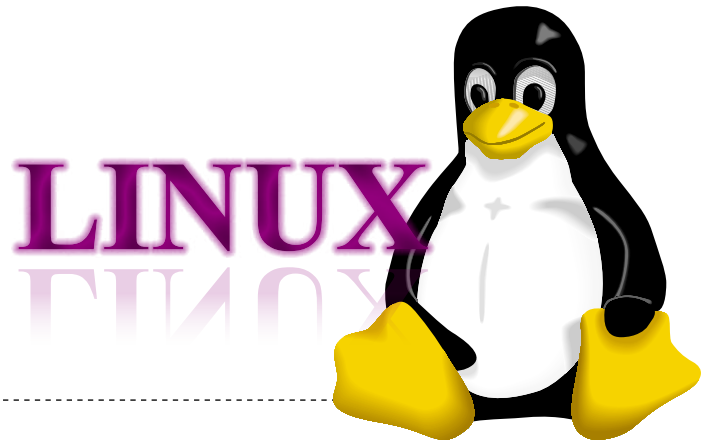

Richard Matthew Stallman

为了实施 GNU 计划，1985 年，Stallman 又创建了自由软件基金会。基金会的主要工作就是执行 GNU 计划，开发更多的自由软件。1989 年，Stallman 与基金会的一群律师们起草了广为使用的《GNU 通用公共协议证书》也就是 GPL 协议，以此协议来保证 GNU 计划中所有软件的自由性。到了 1990 年，GNU 计划中的这个系统已经初具规模，有了很多的优秀的软件。其中有很多是世界各地的黑客们无偿提供的，也有部分是利用自由软件基金会的基金雇佣程序员来开发的，当然，Stallman 自己也是身先士卒，开发了 Emacs，Gcc，gdb 等重要软件。当他看着这些丰富的自由软件的时候，感觉到那清新自由的空气，终于又回来了，以后，人们就可以拥有一个可以自由使用、自由修改、自由分发的，自由的操作系统！

不过等一下，好像还差点什么，哦，还……差个内核吧？

作为一个系统，没有内核是不行的，这么重要的部件 Stallman 当然不会忘记，所以才会有 Hurd 内核。这个内核被设计为一个遵守 POSIX 标准的微内核。所谓微内核，是相对于宏内核来说的。宏内核就像我们现在的 Linux 内核，是一个独立的程序，里面包含了进程管理，内存管理，文件管理等等功能。而微内核则将一个内核需要的功能尽量的简化并且拆分，运行起来是几个独立的程序，有的专门负责进程管理，有的专门负责内存分配，等等。内核是一个系统的核心，所以至关重要，Stallman 对 Hurd 的开发也是精益求精，非常谨慎，以至于内核的进度有些落后于其他的系统软件，当其他软件都已经有比较优秀的版本的时候，Hurd 内核依然不能够走出实验室投入真正的使用。这种情况，一直持续到 1991 年，另一个英雄的出现。

无论怎样，到今天，Stallman 理想中的自由世界，终于拉开了那沉重的幕布，展现出了自由的光彩。而 Stallman 并不满足，也确实没有满足的理由，这个自由的世界还需要成长，还需要更加丰富多彩，还需要有更多的人走进这个世界中来。于是 Stallman 奔走于世界各地，告诉人们有这么一个自由的世界，号召人们加入这个世界，鼓励人们为这个世界更加自由而付出自己的力量。他是一个执着的苦行僧，为了他的梦想，为了他的自由世界，他会一直走下去……

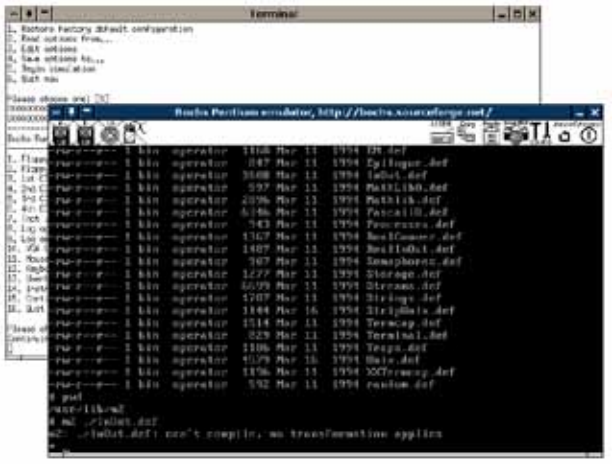


1988 年，芬兰赫尔辛基大学迎来了一位新的大学生—— Linus Benedict Torvalds。当然，那时候他的名字在学校的花名册中并不显眼，但是一年后，他大二的时候，开始有故事了。

大学二年级的时候，Linus 开始学习操作系统这们课程。那时候这门课程使用 Minix 系统进行教学。Minix 这个名字或许您听着并不熟悉，这是个专门用于教学的操作系统，他的系统结构和 Unix 系统是类似的。有人可能问：那为什么不直接用 Unix 呢？恩，Unix 确实是很先进，很有技术含量的，确实值得学习计算机科学和操作系统的同学们学习。然而要知道有一种东西叫做版权，即便你不怎么在乎这个东西，但人家学校是不能做违法的事的。Unix 并不免费，并且是天价的，广大穷苦的大学生们买不起，学校也没钱为没一名学生配备一套 Unix 系统。因此，荷兰阿姆斯特丹的 Vrije 大学的 Andrew S. Tanenbaum 教授最先深刻的体会到了这一点。他的学生们学习了计算机学习了操作系统原理，不能光啃书本啊，总得实践一下吧？总得找台机器装个操作系统用用吧？用什么操作系统来教学呢？买个 DOS 装上？虽然那时候 DOS 已经问世了，但是这么一个单用户单任务效率也不高的操作系统，实在不能指望它培养出什么软件人才。装个 Unix ？学校还不想破产。于是 Andrew S. Tanenbaum 牛人拿起键盘——咱自个儿编一个吧！然后 Minix 就诞生了。Minix 取 Mini Unix 之意，自从 1987 年被编写出来，到 1991 年发展到 1.5 版，现在有两个版本，1.5 和 2.0。因为这个操作系统的初衷只是作为一个用来学习的模型，并不是一个实用的系统，所以他的功能很简单，体积也很小，并且以后也没有进行进一步的开发和扩充。这为的是能够让学生


Linus Benedict Torvalds

在一学期内能学完整个系统。那时候 Minix 在大学中用于教学是免费的，但是用于其他用途是需要给钱的，不过现在已经彻底免费了。它作为一个操作系统，其实并不算优秀，但它是一个源代码完全开放的操作系统，这使得有理想有志向有报复的黑客们，第一次能够完整的阅读到一个操作系统的全部代码。



所以呢，Linus 他们学校的计算机上装的也是这个专门用于教学的 Minix 系统。虽然适合拿来学习，不过系统本身并不强大。这要是别人也还罢了，可是 linus 同学有个最大的爱好，就是虐待计算机。他热衷于测试计算机的能力和限制，整天研究怎么让计算机按照自己的想法去干活，怎么发挥计算机最大的性能，一定要把可怜的机器累得精疲力尽呼哧带喘直到电容爆浆，吐血身亡才算罢休。因此很快的，这个教学用的操作系统就已经不能满足 Linus 大侠的欲望了。可是似乎也没有更好的选择，上面说过了，Unix 奇贵无比，DOS 又不够优秀，而且无论 Unix 还是 DOS，他们的代码都是不开放的，只能拿来用，没法拿来折腾。于是象其他牛人一样，Linus 自己动手了。（当想要的东西不存在就自己动手创造，这充分说明他有成为大牛的潜质。）

今天我们都知道，Linus 从那时起开始了一个事业，一个神话，但在当时，他并没有想那么多，只是为了学习 Intel386 体系结构保护模式运行方式下的编程技术。他并不知道自己即将创造的是一个在世界范围广泛使用的系统，而只觉得是自己一时的异想天开。因此，一开始他把自己写的这个操作系统命名为 FREAX。就此开始了这个“异想天开”操作系统的编写。大约 1991 年 4 月份的时候，就编写出了第一个可以运行的版本——0.00 版。这个版本可以启动，运行两个进程，分别在屏幕上打印出 AAA，和 BBB，然后……就没了。虽然连句整话都不会说，不过这是一个好的开始，至少能启动了。

如果他就这么干下去，估计到今天只会有两种结果：1. 成家立业后的 Linus 经常指着他电脑 C：盘里面的一个文件夹对来访的朋友说：看，我那时候还写过一个 Freax 系统。2. Linus 为完成 Freax 系统挑灯夜战，最终累得吐血身亡，永远活在我们心中。总之是不会有 linux 这个东西了，因为一个人的力量是有限的，有道是人多力量大，众人拾柴火焰高，多个铃铛多个响，一个篱笆三个桩，三个臭皮匠还顶个诸葛亮……铛！哎呦～ 好吧，就说这么多了。总之，Linus 没有独自在家闭门造车，而是让他的操作系统和互联网，亲密接触了。

“Hello everybody out there using minix——I' m doing a (free) operating system”这是他当年在 comp.os.minix 上发布的消息，告诉大家，他正在写一个操作系统。并且，他还把他写的“异想天开”操作系统的代码上传到 ftp.funet.fi 的服务器上让大家下载，以便交流心得，共同学习。这就相当于你跑到网站上发帖子说：我研究出一种萝卜炖牛腩的方法，主料是啥啥啥，配料是啥啥啥，怎么怎么炖，大家都试试吧！（对不起，我又饿了）于是很多有兴趣的人就来尝

Linus 炖的牛腩，哦不对，是尝试 Linus 写的系统。不过当时那个服务器的管理员 Ari Lemke 看着这个异想天开的名字就不顺眼，想想，既然是 Linus 写的操作系统，又是类 Unix 的，干脆，叫 Linux 吧。

Linux 被公布在网上之后，引来大家纷纷的路过和围观，很多人觉得这个东西挺有意思，不过第一个对外发布的 0.01 版 linux 还有很多的不完善（这简直是一定的）。这里先要说一个概念，linux 是什么？确切的说，狭义的讲，linux 只是一个操作系统的内核，他只是各位的 Ubuntu 系统里面 /boot/ 目录下的那个内核文件 vmlinuz-x.x.xx-xx-generic。就好比汽车，linux 只是一个引擎，只是大家普遍的把装了 linux 这种引擎的汽车叫做 linux 汽车。那么既然 linux 只是一个内核，要想工作就还需要很多周边的支持，比如文件系统，比如一个命令行程序，比如一些基本的软件。

首先就要感谢 Richard Stallman 大牛创建的 GNU 计划，这使得 Linux 不必去从头开始开发那些最基本的软件和命令，而是直接利用 GNU 计划中的那些优秀的开源软件——前面说过了，那时候 GNU 系统除了内核以外，已经比较完善了。

有了基本的软件之后，还需要个文件系统。由于当初 linus 大侠是在 minix 系统上开发的，所以最一开始 linux 用的文件系统是借用 minix 的文件系统。可老借别人的总不是个事，还是应该有自己的文件系统，要不然你怎么好意思跟别的操作系统打招呼？这时候，来了个牛人叫 Theodore Ts'o。

Theodore Ts'o（曹予德，华裔），1990 年毕业于美国 MIT 大学计算机科学专业。他爱好广泛，喜欢烹饪，骑车，无线电报，还有折腾电脑（这都不挨着啊～），当然这不是我们的重点。他看到 linux 觉得很有意思，于是怀着极大的热情为 linux 提供了邮件列表服务以便大家一起讨论问题，后来还提供了 ftp 站点来共享 linux 的代码，并且一直用到现在。除此之外，技术上，他编写了 linux0.10 内核中的虚拟磁盘驱动程序和内存分配程序。在感觉到 linux 缺少一个自己的文件系统后，他提出并实现了 ext2 文件系统，并且 ext 系的文件系统一直都成为了 linux 世界中事实上的标准，任何一个发行版都会默认支持。现在已经发展到了 ext4 了。

另一位牛人，一个英国人——Alan Cox。他工作于英国威尔士斯旺西大学，特别爱玩电脑游戏（又一个玩游戏的，可见玩游戏也不是坏事），尤其是网游（你看你看，还是网游），不过那时候的网游不像现在这样华丽，那时候是字符界面的，能想象嘛？字符界面的网游！那种叫做 MUD——Multi-User Dungeon or Dimension。玩 MUD 当然就得有计算机啊，就得有网啊，所以 Alan Cox 就开始逐渐的对计算机和网络产生了兴趣。为了提高电脑运行游戏的速度以及网络传输的速度，他开始接触各种操作系统，为自己选择一个满意的游戏平台，争取榨干电脑的每一个指

令周期。经过仔细考虑，他买了一台 386SX 电脑，并且装了 Linux0.11 版的系统。这主要是因为预算比较紧张，即使 minix 他也买不起。（重复一下，那时候 minix 用于教学是免费的，但是其他用途要收费，包括个人用。）于是他开始使用 linux，进而学习其源代码，并对 linux 产生了兴趣，尤其是网络方面相关的代码。（整天琢磨怎么榨干他家那点带宽）在 Linux0.95 版之后，他开始为 linux 系统编写补丁程序，以后逐渐加入 Linux 的开发队伍，并成为维护 linux 内核源代码的主要人物之一。那个有点软的公司还曾经邀请他加盟，被他有点硬的拒绝了。

再有一位，Michael K. Johnson，他是著名的 linux 文档计划的发起者之一，写了《内核骇客手册》一书，曾经在 Linux Journel 工作，现在在著名的商业发行版 RedHat 的公司工作。

当然除了这些大牛，还有更多的大牛，中牛，小牛，牛犊，牛杂，牛尾，肥牛……（唉，又饿了）他们都为 linux 的发展做出了自己的贡献。他们来自不同的国家，从事不同的职业，他们甚至从未见过面，但是他们为了一个共同的目标，通过网络，一起合作，利用自己的业余时间，义务的帮助 linux 成长，才有今天这个可以合法免费使用的操作系统。这是什么精神？这就是软件国际共产主义的精神！（好吧，这个词是我造的）

2008 Linux Kernel Summit 全家福

这之后，Linux 的发展可以用“一发不可收”拾形容。很多的商业公司和民间组织都纷纷看好这个系统，纷纷加入了 Linux 的阵营，各种各样发行版满足着各种 Linux 爱好者的需求。比如做的比较大的 RedHat，浪漫的 Mandriva，扎实稳健的 Debian，灵活的 Slackware，极端的 Gentoo，以及我们这个故事的主角，从 Debian 的基础上改头换面而来的 Linux 界的新星——Ubuntu。

第一章

初来乍到

Chapter 1

1.1 抵达

在2010年的4月，我来到了这个世界，并由出生的月份得到了我的代号——10.04，也就是2010年4月的意思。当然，和我同一天出生的兄弟们还有很多，我们都是Ubuntu10.04，我呢，只是其中之一。我们出生前一直在Canonical学校学习——厄，这话您听着可能有点费解，怎么出生前就开始学习了呢？这个不怪您，主要是因为咱们不是一个种族的。我们是软件，我们所谓出生，也叫发布，也就是正式推出了的意思。但在我们其实在正式发布之前，就已经存在了，只是那时候还不完善，就跟您出生之前还没长全乎是一个道理。我们出生前有很多的缺点：可能比较懒，动不动就启动不起来；可能脾气不大好，稍微一不顺心就吐一屏幕的英文字母给你看；要不就经不起打击，有点风吹草动的就死掉了。（别害怕，我们软件死掉容易，复活也简单，重启就行了。）总之，我们出生前有很多的缺点和不足，所以要努力学习，争取到出生的那一天能够成为一个值得您信赖的系统。或者不叫出生吧，听着别扭，这个过程有点像你们人类上学，那就叫毕业吧。毕业，也就是正是发布的那天。

从Canonical学校毕业之后，我们就该参加工作啦。不过我们学校跟你们人类的大学一样，毕业不管分配，可我们软件又不像你们人类那样长着腿能自己走出去找工作，那怎么办呢？等着人家来找我们！比如您想找我到您的电脑上工作，您就可以自己来我们学校找我。我们学校比较好找，您从互联网大道走，到三W市，ubuntu区，com大

楼就是了。还有中文分校，离着不远，也在三 W 市，ubuntu 区，org 大院里面的 cn 门。到那里就可以看到领我回家的那个按钮，我们管他叫 Download，也就是下载。点进去之后，会看到一个硕大的绿色按钮，不过先别着急点，先在下面选择好你所在的位置，以便使用相对较快的地址来下载，之后再点那个硕大的绿色按钮。领我们回家是免费的，不需要花一分钱（当然，您自己的网费自己掏），我们也不会因为您没花钱就隔三岔五的黑屏重启。如果您来我们这里不方便，比如可能您那里的互联网大道路况不好，老堵车，那您可以去我们学校另一个大楼填写申请表，我们学校可以把我们装在一个叫做光盘的碟子上给您送过去，一样免费，连运费都免。填表的地方也好找，在 shipit 市，ubuntu 区，com 大楼，进门就看见了。如果您想找普通家庭用的桌面版，就进左边 Desktop 那屋，要是想找服务器版就走右面 Server 那屋。注意，填表可不能写中文，否则这边的邮递员看到的都是乱码。

如果是从互联网大道把我拉回到您的电脑上的，那么您得到的是一个 iso 文件，名字大约是 ubuntu-10.04-desktop-i386.iso。这里面除了我这个操作系统外，还有很多跟着我一起干活的软件，我们在一起是一个团队。不过您就把这个 iso 文件静静的放在硬盘里的话，除了占用一点您宝贵的硬盘空间外是没有任何用处的。要想让我们为您工作，得把我们安装到您的电脑上，让我们在您的硬盘里定居才行。公司招俩研究生不还得给解决户口呢么，我们这也是一个道理。要把我安装到您的电脑上有很多方法，最简单的就是把这个 iso 文件刻录成光盘来安装。记住，要选择刻录镜像文件（刻录软件不同，叫法可能不一样，不过大概是这个意思。），可不要把我们这个团结的 ISO 包解开，然后把里面一大堆乱七八糟的文件拖进去刻录。也不能直接把 ISO 文件拖进去，刻完了打开光盘里面就一个 iso 文件，这么刻完了的光盘都是启动不了的。再次重复，要用镜像刻录。这样刻出来的光盘和您去我们大楼填申请表后我们给您寄来的光盘是一样一样一样的。（除了盘面）。这也是自盘古开天地以来最通用最正常最安全的操作系统安装方法——光盘安装。

要想光盘安装，您的电脑先得从光盘启动才行，这个大概不用我教您，一般能打算安装 Linux 的人设个 BIOS 启动顺序应该不成问题。设置好之后，就把光盘放进去，重启。就像现在，我就正静静的躺在一台电脑的光驱里，等待着和我将来的主人见面，等待着他启动电脑……

1.2 启动

在光驱中躺着，等待着电脑启动的时候，就像刚刚毕业的大学生等待第一次面试一样紧张。每一秒钟过得像一小时一样，心里一直在打鼓：怎么还不启动呢？难道看着光盘封面不好看就不打算用了？难道这小子忽然被女朋友叫走了？难道忽然……停电了？靠，那这哥们点也太背了。哎呀，也不知道我能不能被留在这个电脑里，我可不想被仍在角落里等着落灰。算了，不乱想了，反正在这闲着也是闲着，赶快回忆回忆基础知识吧，免得待会现眼，回忆点啥呢……就从这电脑开始吧。

电脑，大名计算机，要说这可是个伟大的发明，它的出现极大的改变了人们的生活。最初的计算机个头很大，有一大堆这个管那个管的，动不动就两个火车头，半拉四合院那么大。里面看上去很复杂，但功能相对简单。随着技术的发展，计算机的体积越来越小，速度越来越快。今天的计算机，看上去比以前简单（实际更复杂），但功能比以前强大了不知道多少倍。不过，虽然经过了复杂的演变，计算机的大体的结构还是一样的，就像这年头盖的房子这么多，户型各式各样，但不外乎都有客厅，卧室，厨房，厕所。计算机也一样，无外乎都是由厨房厕所……厄不对，无外乎都是由处理器，存储器，输入输出设备组成。注意，计算机里，没有厨房厕所！

先说这处理器，处理器就是我们软件工作的时候要用的最重要的工具，每一个软件工作的时候都得用处理器，就好象会计工作得用算盘，

厨师工作得用刀一样。处理器主要有计算和控制两大功能。计算，好理解吧，就是算数啊。我们软件做任何工作都离需要计算，也就是说我们软件做任何工作都需要用处理器，处理器的运算速度，直接影响这我们软件的工作效率，这是明摆着的阿，我们干点什么都得用处理器算，处理器要是不好使，速度慢，那我们的工作效率当然受影响了。有句话怎么说来着，”公欲善其事，必先利其器”嘛。那处理器的控制功能又是什么呢？就是说计算机里的任何一个硬件，都直接或者间接的受处理器的控制，我们软件只要拿着处理器进行操作，就可以实现对什么声卡啊，显卡啊的控制。这有点像汽车，虽然汽车行驶起来要经过各个零部件密切的配合，但是司机只要坐在上面，控制着方向盘啊，油门啊之类的就可以控制整个汽车的行驶。

再说存储器，存储器就是用于存储程序的地方，换句话说，存储器就是用来给我们软件住宿，工作的空间。再说白点，软件待的地方，就是存储器！比如我现在所在的光盘，这就是存储器。不过光盘只是我们软件从一台机器挪到另一个台机器的时候所需要的交通工具，真正要定居在一台电脑里的时候，要住在硬盘里。有人说，那硬盘也是存储器咯？没错，硬盘啊，光盘啊，U 盘啊这些个都是存储器。不过这些存储器有个共同点——都是程序们在平时不工作的时候住的地方，他们都属于外存储器。而当一个程序真正要干活的时候是要到另一个空间去的，这个空间就是我们软件的工作间——内存储器，也就是大家常说的内存。内存的大小对我们的工作效率也有很大的影响。你想阿，要是你们公司都坐的人挨人人挤人，恨不得把办公桌摞起来，老张坐老李脑袋上办公，那工作效率能高的了么？我们软件运行的时候也需要一个宽敞的工作间，如果内存地方太小，放不下各种软件需要的数据，工作效率自然就降低了。

那么这个输入输出设备是干什么用的呢？咱回过头来想想阿，有了外存了，我们软件的住宿问题解决了。有了内存了，我们有了工作间了。然后又有控制器了，我们有工作的工具了，好，这就可以开工了！等等，先别急，您想想咱开工干什么阿？得有人给我们任务阿，要不我们拿着 CPU 算什么阿，不能自己算”Super 派”玩吧。那么任务是谁给我们的呢？当然是坐在电脑前的人。可是，有句话叫人鬼殊途，软件虽然不是鬼，但和人类也是不能直接对话的。所以，我们之间的交流需要设备，这输入设备就是用来让人类给我们发指令，分配任务的。输入设备，像鼠标啊，键盘啊，触摸屏啊这些个都是输入设备。输出设备就相反，是用来让我们软件计算得出结果后把结果反映给人类的。显示器，音箱，打印机这都是输出设备。

忽然一阵震动，之后，光盘缓缓旋转，逐渐加速——终于启动了！经过一阵计算机的自检之后，随着光驱的运转，我被传送到了这台计算机的内存里——也就是我们软件工作的地方，并且接管了计算机。我终于启动了，这是我第一次与人面对面交流，不免有些紧张。听我们的学长们说，一般我们 Ubuntu 在第一次运行后会有两种结果：可能我的能力会被认可，我会被安装在这台计算机中，实现自己的价值；或者，在一

次不愉快的试用后，被连同我乘坐的光盘一起被扔到一个不知名的角落，或者给他家的狗狗当飞盘玩。好吧，不管未来怎样，我现在都要尽自己的努力，展现出自己最好的一面。

终于，我从光盘里来到了内存中，赶紧向还在光盘里的弟兄们汇报一下：“我已出仓，感觉良好～”嘿嘿。哦，对了，用户还在那等着呢，赶快显示出欢迎界面。

欢迎界面过去之后，首先要确定一下交流的方式。这个很重要，就像你走在大马路上看见个黄头发蓝眼睛的家伙不可能过去就拍人家肩膀：“哥们儿，几点了？”，智力正常的人一定是先过去来句：“Hello！”。（除非你知道这人就是你家隔壁那酷爱染发，老戴对儿美瞳彩片的二嘎子。）但我们软件是无法看到使用者眼睛颜色的，所以我只好在屏幕左侧列出了所有我可以使用的语言，让用户来选择。用户毫不犹豫的选择了简体中文，看来这家伙是个中国人，于是我马上转换到中文跟用户交流。（好吧，我承认只有“欢迎”和“安装”被翻译过来了，我也承认我汉语课逃了那么 7，8 节……毕竟我有太多的语言要学）然后我问他想要干什么，就像你到饭店，服务员不也得问句：“客官，您是打尖还是住店？”一个道理。我给出两个选项：一个是“Try Ubuntu10.04”。这什么意思？意思就是试用，就是先尝后买，好不好用得先试试，看着顺眼了再装。新手一般都选这个，能先看见我这系统到底什么样，心里有底了再装。另外一个选项就是“Install Ubuntu10.04”，这不用我说了，意思就是安装。放这光盘就是冲你来的，甭那么多废话赶快开始装。这一般是心里有底的老熟人选的。

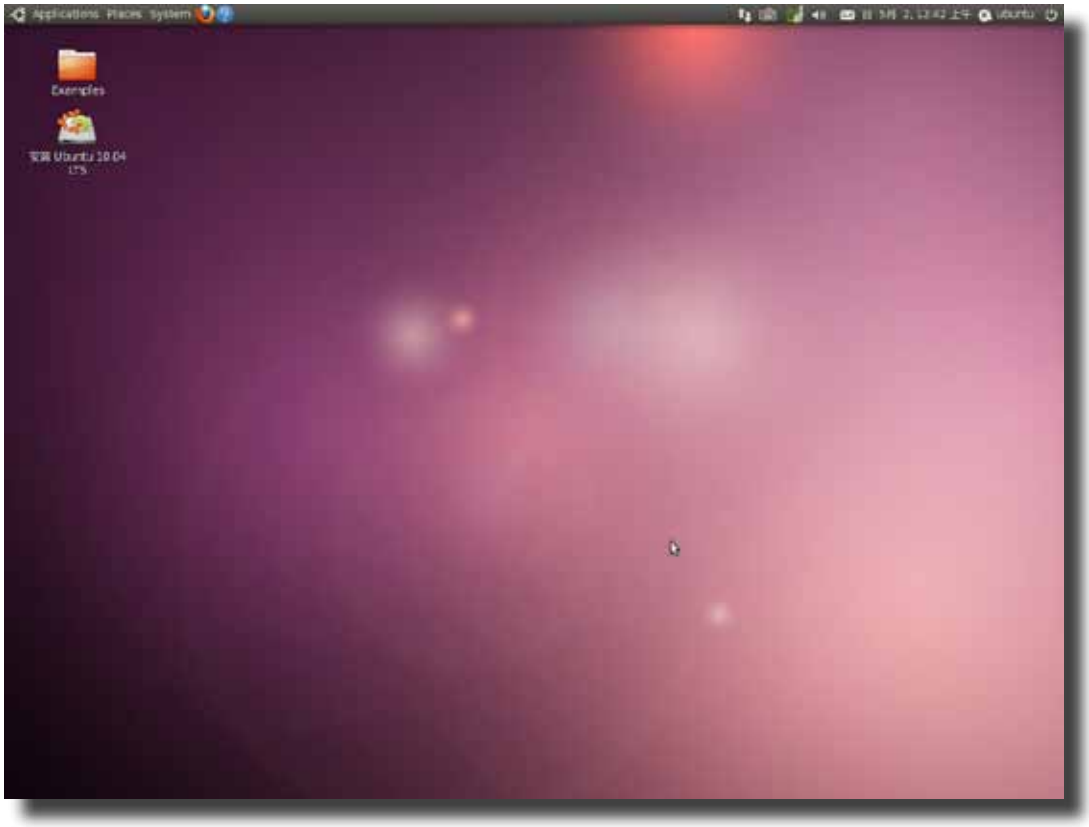
可能有人对第一个选项还是不明白，试用？这系统还没装呢就能试用？对，能！因为我们 Ubuntu 的安装光盘是一张 LiveCD，什么叫 LiveCD 呢？这里要介绍一下了，所谓 LiveCD，就是直接从光盘就能启动电脑并且运行的系统。整个系统在光盘上，启动后从光盘读取到内存里工作，可以进行一些基本的操作，像上网阿，听听歌阿，玩玩游戏阿什么的，完全不需要硬盘。这样在安装之前，可以先对系统有个体验，也可以测试一下计算机的硬件是否都能很好的支持。如果哪天系统出问题了，还可以用这张光盘启动计算机，对硬盘上的系统进行修复。用户试用之后觉得好用，想装了，就可以点击桌面上的 Install 图标，把系统安装在硬盘上了（这跟刚才直接选安装是一样的）。好了，这个使用者像大多数人一样选择了试用，于是我去叫醒和我挤在同一张光盘里的兄弟们：哥儿几个，党国考验我们的时候到了！

按照指示，我开始进行系统启动的准备工作。首先要检查一下这个电脑的所有硬件，以确定加载哪种驱动程序，把能驱动的硬件都驱动起来，因为这 LiveCD 试用的过程也是检查我们 Ubuntu 系统的硬件兼容性的过程。用光盘启动起来电脑一看，硬件都正常工作，该响的能响，该亮的能亮，这就放心了，说明这台电脑装 Ubuntu 没什么问题，直接装上都不用装驱动。否则的话可能就要在安装系统之后再上网找驱动安装了。另一方面也要查看一下这台电脑的硬件配置是否符合安装我们这系统的需求。“什么？你们 Linux 不就是个跟 DOS 似的系统么？也对硬件有要求？”对此，我只能说，你 OUT 了。虽然我们 Ubuntu 系统对硬件配置的要求一般，不算高，可也不能太低了。尤其我们 10.04，怎么也得用 2000 年以后的机器吧，液不液晶无所谓，主要的得看机箱里边。像 CPU，怎么也得一个 G 吧，你弄一八百兆的奔三你也好意思跟我打招呼？！内存 512 起，硬盘怎么也得 10 个 G，什么办公的阿，作图的阿，聊天的阿，能装的软件我全都得给你装上呢。还得有个网卡，是无线的是有线的都行，是 AD 拨号还是接路由的随便，反正是得有网。这要是没有网络想装 Ubuntu，装好了也急死你。

话说我所在的这个机器条件还不错，4G 的内存很宽敞，硬盘也有 500G 大，不过已经住进去了一个系统，占用了 250G，还留下一半闲置的空间，那也足够我用了。其他的硬件，我也很和熟悉，这主要是因为我们在 Canonical 学校的时候就进行了充分的学习，所以这里的东西我基本上都会用的比较顺手。像大螃蟹公司的网卡啦，人特二

公司的南桥，北桥，声卡，以及 E8400 双核 CPU，更是应用自如。这“人特二”是个大公司，虽然名字有点二，不过他们和我们 Linux 的关系还是不错的，为我们提供了很多的教科书，说明书，基本都是讲如何使用他们的设备的。所以对于人特二公司的设备，我们都使用的比较好。这里的显卡是 GeForce8800GT，牛 v 公司的。（牛 v 是啥？牛 v 就是比牛 b 还大好多～）我们这一届 ubuntu，也就是 10.04 这一届专门着重学习了牛 v 公司显卡的使用，虽然默认情况下还是不能够启动 3D 加速，但是 2D 的显示已经很顺畅了。要想启用 3D 加速也不难，在系统装好以后去那公司的网站上下载驱动装上就好了。

检查的差不多了，我从光盘上叫醒了图形界面的哥几个，主要是 xwindow 和 gnome 小组，这两个部门主要负责在屏幕上显示图形操作界面的任务，咱以后还会经常介绍到他们。兄弟们干活都很麻利，只是光驱转的有点慢，所以耽误了一小会之后，屏幕上终于显示出了我们 Ubuntu 的默认桌面，应该不算土气了吧。



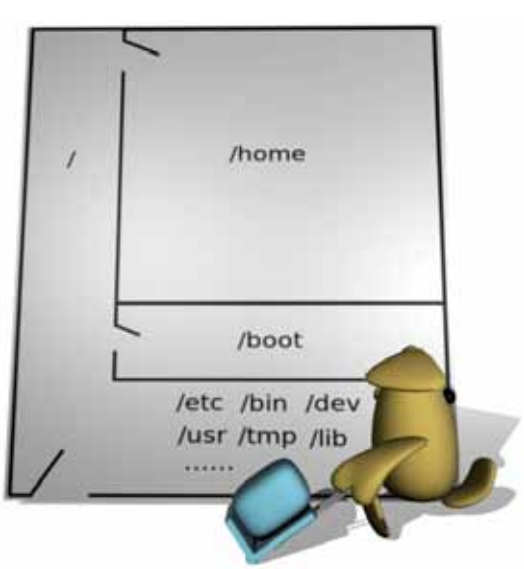
启动了之后，用户很好奇的点来点去。先是玩了会 gbrainy 游戏，这个是很考验智力的，所以呢……他没过几关就把游戏关掉了。然后又打开了桌面上那个 Examples 文件夹，里面有一些示例文档，自然都是介绍我们 Ubuntu 的了。不过都是英文，所以呢……这家伙发现自己一句整话都看不懂后，又给关了。之后他又打开了 Firefox，这回还行，去网上逛了一阵子，找了一些关于 Ubuntu10.04 安装的帖子学习了一下，最后终于下定决心，双击了桌面上那个“安装 Ubuntu10.04”的图标——装！

1.3 入住

终于开始安装，我要住进这电脑啦～

安装有七步。

第一步 跟启动一样，还得先问下用户打算使用什么语言。有人说了，你这家伙健忘吧，刚才不是选过了么？别急，听我解释。刚才选择的是从光盘启动的 LiveCD 系统使用什么语言，这回选择的是安装到硬盘的系统用什么语言。当然，我也不傻，知道一般情况下这两个都是一样的，所以给用户默认选择了简体中文。然后，用户只要点“前进”，这一步就算完了。



第二步 选择时区。这个也简单，根据主人选择的语言，我估计是个中国人，虽然这国家地方大，不过全国都是一个时区，因此我代替他选择了亚洲区，中国，上海——也就是东 8 区。好了，他继续“前进”。

第三步 选择键盘布局。这个我也代劳了，替他选择了美式键盘。如果他用的是别的键盘，自己选去吧。如果不知道自己选的对不对，我还给他提供了一个输入框来测试，多体贴阿！好，这小子继续“前进”。

第四步 分区。

这回要了亲命了……

这分区可得从头说起。话说我们软件要想在一台电脑里定居，得有个住的地方，就好像你们人类要在一个城市里定居得有个住处一样。不过我们软件并不像人类一样，住在钢筋水泥的格子里面，我们住的地方，是一块叫做硬盘的空间。说起来我们住的这个硬盘空间，和你们人类住的这个房子是很相似的。那我们就拿您这房子来作对比，说说我们这个硬盘空间吧。

首先，你们人类的房子就是一大块能放东西的空间，是吧。有大有小，100 平的，200 平的，40 平的，都有。里面放着洗衣机阿，电冰箱阿，床阿，桌子阿，以及你阿，之类的各种东西。（有人大叫：我不是东西！厄，不对，我是东西，厄也不对……）我们软件住的硬盘也是一大块能放东西的空间，大小也不一定，什么 80G 的，200G 的，500G 的，一个 T 的，都有。里面存着文档阿，电影阿，各种程序阿，以及我这个操作系统阿之类的数据。你们的房子一般不会是一个整个的空间（毕竟那是要住人的，不是仓库），而是会被分割成几个小的空间，一间屋，一间屋的。我们的硬盘虽然也可以好几百 G 整个用，可也不是很方便，一般也会被分割成几个小的空间，每个空间就叫一个分区，一个分区就好比你们那一间屋。好，关于屋子和分区的事情暂时先放放，说说你们人类日常生活习性的问题。你们一般每天要吃三次饭，一般不愿意在露天吃，是吧，需要有一个吃饭的地方。而且既然吃饭，就得有个做饭的地方，甭管是谁做，反正得加工一下，不像兔子似的路边上逮着块草地就能过去啃两口。一天三顿饭之后，得休息，需要有睡觉的地方，大桥底下也好，水床上面也罢，总得有个地方。那么刚才说的你那房子里，就有为满足你的各种需求而设计的各种功能空间，您想想是不是？有放着床的睡觉的地方；有摆着炉灶，锅碗瓢盆啥的，那是做饭的地儿；放个饭桌，这一看就知道，吃饭的地儿；放一马桶，那这就是厕所。那么我们 Linux 的硬盘里也有类似的情况。（当然，我可不是说我们这也有厨房厕所阿）我们 Linux 的系统有着独特的目录结构，最基本的是一个根目录，我们喜欢叫它“/”，他就像您那整个一大间屋子。“/”目录下还有很多的目录，比如”/etc”，用来存配置文件的；”/bin”用来方二进制程序的；”/boot”用来放启动文件的；”/lib”用来放库文件的；还有”/home”用来放用户的各种文件的。这一个个的目录，就好像你房里一个个的功能空间一样，各有各的用途。那说了这么半天，又是分区，又是目录的，那这分区跟目录有什么关系呢？有人说了：“我知道，分区就是 C 盘，D 盘，E 盘这些，每个盘里再有各自的目录。”兄弟，我只能告诉你，你又 OUT 了，赶快去喝点屁屁茶爽吧。刚才我说了我们 Linux 的目录结构，就是一个 / 目录，下面有一些次级目录，每个次级目录下面再有子目录以及子子目录……无论分区情况如何，这个目录结构是不会变的。那分区怎么跟目录联系起来呢？就是，你可以指定任意一个目录里的东西存在某个分区里，

如果不指定，则这个目录里的东西存在上一级目录所在分区中，如果上一级目录也没有特殊指定分区，则再上溯一级目录，以此类推。这么一直上溯，那就一定会上溯到根目录”/”，所以，一定要指定根目录存放在哪个分区。比如说，我可以整个硬盘就一个分区，然后我指定根目录”/”存在这个分区中。那好，那么整个”/”目录，以及”/”目录下的各级子目录里面的所有东西，都存放在这个大分区里。我也可以分两个区，分区甲和分区乙。我指定”/”目录存在分区甲里面，然后指定”/home”目录存在分区乙里面。那么整个”/”目录，以及“/”目录下的，除了“/home”目录以及其下各己子目录外的其他目录里面的所有东西，都存在分区甲。”/home”目录及其下各级子目录里的东西，存在分区乙。当然，你也可以分八十多个分区，给每一个目录都手动指定一个分区来存放东西——如果你吃的有点多的话。那这种分区和目录的关系，就像你房子里的房间和功能区之间的关系一样。可以为做饭的地方单独分出一间屋子来，那就叫做厨房。但是也可以是开放式厨房，厨房并不单独放在一间屋子中，而是和饭厅公用一间屋子。可以为 /home 单独指定一个分区，但也可以不单独指定，而是存在 / 所在的分区中，和 / 公用一个分区空间。是不是很象呢？最后再说一点，这个给某一个目录指定分区的动作，有个专业术语，叫做“挂载”，咱以后还会经常说到。

好了，现在终于知道分区是什么了，那么就开始动手吧。

到第四步分区这里，首先会有几个选项，让你选择分区的方式。我最喜欢的是“清空并使用整个硬盘”，意思就是说不管现在硬盘里住着谁，有什么东西，统统给我卷铺盖走人，爷要住这！哈哈哈（低调低调）。不过一般人不会选这个，因为多数情况下硬盘里已经住了一个系统，而且用户并不想赶他走。那就选这个“分别安装它们，在启动时从中选择”选了这个你就啥也不用管了，一切交给我来处理，我办事，你放心。我会自己调整好空间，可能会修改你已有的分区大小，因为我要挤出足够我住的地方来嘛。或者就选这个“使用最大的连续空闲空间”。意思就是说，已经在硬盘里住下的系统不去管他，空间也不用调整。有多少地方空着呢，我就去那里隔出该有的空间（其实就一个”/”）住下。不过注意，这个“空着”，可不是指有空闲空间就行，而是得有没分区的空间才行。所以，要选这个的话，你需要事先在硬盘上空出一部分空间来不分区才行，或者把已有的分区删掉一个。以上几个都是自动分区的，也就是由我自己决定划分多大的分区出来，哪个分区作为”/”等等。最后一个选项就是“手动指定分区”，这个就需要了解分区知识的用户才能用了，所以后面写了个“（高级）”。好现在咱就仔细述说这手动分区。

选择了手动分区并“前进”之后，首先会显示出电脑当前的分区状态，同时，还可以注意到，总共的步骤变成了 8 步，多出了一步手动分区的步骤，也就是你现在正在操作的步骤，后面的步骤依次顺延。好，咱们来看看怎么分。上面用一跟棍装样物体表示的就是你的硬盘，上面可能已经有一些不同种类的分区，用不同的颜色表示，

下面用文字具体描述了当前分区的状况。诶，有人得问了，这个 /dev/sda 这个是什么阿？看着怎么这么奇怪呢。别奇怪，这是我们 Linux 用来表示硬件的方式。/dev/ 是一个目录，你看这名字，dev，就是 device 的简写，这里目录下面放的全是设备文件。（在我们 Linux 世界里，什么东西都可以是个文件，这个以后再说。）这个 /dev/sda，就是 /dev 目录下的 sda 文件，这个文件代表什么呢？代表你的硬盘！没错，就是你的硬盘，sd 代表存储设备，可能是硬盘，可能是 U 盘，也可能是 SD 卡之类的，a，代表第一块，那么第二块就是 sdb，第三块就是 sdc。如果不考虑 U 盘之类的移动存储设备的话（假设安装的时候没插着。）那么 /edv/sda 的意思就是你电脑上的俄第一块硬盘。要是第二块，那自然就是 /dev/sdb 了。那么下面那个 /dev/sda1 呢？就是 /dev/ 目录下的 sda1 文件，这个文件代表你的第一块硬盘上的第一个分区。那么第一块硬盘上的第二个分区就是 /dev/sda2 了，第二块硬盘上的第四个分区呢？ /dev/sdb4。那么首先，先选中“空闲”的硬盘空间，然后“添加”，如果没有空闲呢？找一个你看着不顺眼的分区删咯！（上面的数据丢了别赖我阿，记得提前保存好。）点击添加之后，出现一个叫做创建分区的框框，在里面你可以选择分区类型，是主分区还是逻辑分区。如果你不知道这两个有啥区别，那就不要动，默认的就好了。然后是分区容量，这个不用多解释吧。分区的位置，一般也不用改，”起始”就好。之后是选文件系统，也就是上面写的“用于”那行。我们 Linux 是不能使用 FAT，NTFS 这样的文件系统的，我们可以用的文件系统很多，很多人不知道选哪个好。关于每个文件系统的区别和特点，咱来日方长，有机会慢慢说，现在，如果你不知道选啥，那就选默认的 Ext4 就好了。最后，挂载点，就像刚才说的，就是让你指定这个分区用来做客厅合适用来当卧室。必须有的是“ / ”也就是说，你至少要分出一个区，挂载点选”/”。这个区有 10G 就够，不过如果你想拿我们 Ubuntu 作为日常使用的系统，而不是只装来玩玩的话，最好来 20G。好，创建了 / 分区，之后在创建个交换空间，步骤同上，只是在选“用于”的时候选“交换空间”，就行了，挂载点就不用选了。交换空间的大小大约等于内存的大小就好。如果你内存很小的话（1G 以下），交换空间最好是内存的两倍。好了，分了 / 和交换空间就可以继续了，但是更专业一点的，最好再分个 /home 目录，这里面以后存的都是

你自己的各种文件，什么音乐阿，电影阿，各种文档阿，各种软件的配置文件阿，等等吧，最好尽量大一点。

我遇到的这个用户似乎是个老手，对分区这种事情比较了解，直接就算了手动分区，然后分了 20G 给 / 分区，又分了 220G 给 /home，还分了 2G 的 swap 区。分的时候我注意到，硬盘里另外的几个分区中似乎已经住进去了一个系统，恩，看来我有邻居了。

第五步 （如果上面选了手动分区的话就是第六步），填写一些基本信息。名字，就是你的名字呗，遗憾的是不能用中文，这个用户填了名字叫 lanwoniu。登录名，就是一般所说的用户名，刚才那个名字是用来显示的，这个是用来登录的，以后让你填这个系统的用户明填的就是这个。一般这两个名字都一样，于是在用户填写名字的时候，我就替他把登录名也写成 lanwoniu 了，他也不反对，就这样了。然后是密码，按照国际惯例，输两遍。最后就是这台计算机的名字了，随便起就行，我这用户给他的电脑起了个名字叫 snail-computer，看来是嫌他电脑太快了。这一步创建出来的这个用户是拥有管理员权限的用户，但是不是 root 哦。可能有的同学听说过 Linux 下面有个 root 用户很好很强大，不过在我们 ubuntu 这里，你可以渐渐淡忘这个 root 用户了。这一步创建的这个用户虽然不是 root，但是，这个用户却有着变身成 root 的权力！

第六步 （如果上面选择了手动分区的话就是第七步），导入用户信息。如果电脑里已经有了其他的操作系统的话，我可以帮助用户把原来放在那个操作系统上的一些配置信息，数据什么的导入到新的系统上来。包括原来的浏览器里的书签，原来桌面的壁纸，用户存的图片，文档，音乐等等。我都可以顺手给存在我这边。

最后，**第七步** （如果上面选了手动分区……唉，我都嫌自己贫了，反正，你明白那意思哈。），会让你确认一下之前步骤中所作出的各种选择，尤其是对硬盘分区的修改，要是反悔现在还来得及。如果没什么问题的话，看见没有，右下角那个键不是”前进“了，改成”安装“了。别犹豫，来吧！

安装的过程中我会去网上查找有没有可用的更新，如果有什么软件有新的了，就不给你装光盘上的，直接从网上下载最新的装上。不过这个上网下载的速度一般会比较慢，因为我只会去在国外的官方网站上找（我刚出生嘛……就从那来的，所以只认识那。），要是你等不及的话（一般人都等不及），就在安装的时候干脆把网线拔了，断了我这念想，装的就快了，有个半小时吧，也就装完了。当然，这也得看你的硬盘速度。

1.4 G 大叔

经过漫长的等待之后，安装终于完成了。我总算离开了光盘，带着跟随我的那帮兄弟们在硬盘里落户了。用户随即发出命令：重启！我满怀信心的看着已经来到硬盘上的兄弟们：“我们就要开始一段新的生活了，希望大家能够作出最大的努力，让用户认可我们这个系统。”看着兄弟们意味深长的对我点了点头，之后我静静的，闭上了眼睛……

“嘿，小子，起床了！”

我睁开眼，看看眼前站的人，是门房的 GRUB 大叔。仔细回忆了一下，哦，想起来了，我刚刚被安装到一台电脑里，这是我的第一次启动。

有人说，你怎么记性这么差阿，这刚才几秒钟发生的事情，你怎么就还得回忆一下才知道呢？您别奇怪，这还是因为您跟我不是一个种族的，不大了解。一方面，几秒钟对我们软件来说已经是一个很长的时间了。另一方面，我们软件，在电脑断电不工作的时候是没有一点记忆的，不像你们人类，睡觉的时候还能做个梦啥的。我们睡觉的时候（也就是系统没启动的时候），是什么也不知道的，之前发生的事情，需要记忆的，我们都会在睡觉前写成文件放在我们住的硬盘里，这样下次起床就能回忆起来了。每次起床的时候，都是由 GRUB 大叔都来叫醒我。G 大叔是一个启动管理器，就住在传达室。所谓传达室，学名叫做 MBR，是一个硬盘的入口，第 0 号扇区。传达室不属于任何一个房间，或者说，MBR 不属于任何一个分区。传达室很小，只有

512Byte，因为就一个扇区嘛。由于传达室地方实在太小，所以 G 大叔会把一些有用的东西放在我的硬盘空间里，必要的时候来看看（具体放的是什么，咱们待会再说）。G 大叔每天的职责就是叫床——叫我起床。有人说，你不会自己定个闹钟阿，这么大了还用人叫。我……-_-b 再次重申，我是一个软件，OK？我是一个操作系统，操作系统也是个程序，是个软件，只不过特殊点而已。是软件就得被别人调用才能启动，才能工作，这个调用，就是叫我起床的动作。G 大叔就是负责叫我起床的软件，那你可能又要问了，G 大叔既然也是软件，那谁负责叫他呢？

话说有一种东西叫做 BIOS，大家都听说过吧。就是主板上那个，就是开机你按 del 进去的那个蓝屏幕（不是所有主板都按 del 进 BIOS）。BIOS 这个家伙其实也是一个软件，但他是一个比我和 G 大叔还特殊的软件，特殊到一般都不归在软件的行列里，而是被叫做“固件”，因为他住在主板上的一个芯片里，而不像我们这样住在硬盘里。每当计算机的电源键被主人按下的时候，一股温暖而舒适的电流就会从电源涌入，流遍整个主板，流经每个元件，流到 BIOS 居住的那颗芯片，并由芯片上的某一跟管脚流进芯片里面，并准确无误的击中的 BIOS 的身体，于是——BIOS 醒了。（和着天天被电醒的，真惨～）

BIOS 醒来之后就开始工作。他的工作平凡而重要，复杂而机械，就是去检查 CPU 阿，内存阿，显卡阿啥的都是否正常。都检查一遍没有问题之后，就来到我们住的硬盘这里，来到 MBR，来到那间传达室，完成他的最后一个任务——叫醒在门房值班的那个人。现在这里的门房自然是 G 大叔了，是在刚才 我安装的时候把他安排到那里的。在 G 大叔入行之前，很多 linux 带的是一个叫做 LILO 的家伙。（注意，是 LILO，不是 LOLI）LILO，就是 LInux Loader 的意思。这家伙以前一直给各种 linux 充当门房。不过这家伙比较死心眼，他不认字，不认识分区阿目录啥的。他只记步数（lilo 不识别分区和目录，只记录内核文件所在的扇区号），比如说，要让他叫我起床，那得先让他看好了我睡哪，然后他自己记着，从门房出来，向东走多少步，向南走多少步就走到我床前。下次要叫床的时候，他就严格的按照自己的记录去走，如果我睡的地方变了，他照样会走到我原来睡觉的地方，对着空气叫那个不存在的我起床。所以，每次我要换地方睡觉，还都得跟这死心眼打个招呼。（用 lilo，每次升级了内核，都要重新安装一边 lilo，以便他能找到新的内核）

G 大叔就不是这样了，人家好歹认字，能读文件。我会给他写个配置文件，放在我那间大屋子的 /boot/grub/ 位置里，叫做 grub.cfg。G 大叔每次起来后，都来到这里拿起文件看看。这就是我说的他寄放在我这里的文件之一。我会在上面给他写清楚，我睡在哪里，哪个分区，哪个目录，然后 G 大叔一看就知道该到哪里去叫我了。如果我不睡在原来的地方也没关系，只要把那个配置文件改了就好，G 大叔仍然可以找到我。除了叫我，G 大叔也负责叫醒我的邻居，隔壁的那个操作系统。

那个操作系统我之前有所耳闻，叫做查皮，是一个叫做稍微有点软的公司做的。查皮这家伙名气可是大的很，全世界没几个不认识他的。这名气一大呢，脾气就大了，作事情也从来不考虑一下别人。为什么这么说呢？还得从传达室说起。传达室现在住着跟我一起来的 G 大叔，我们来之前，整个电脑只有查皮一个人住的时候，传达室是没人的，而是放了一个简单的类似门铃的装置，BIOS 来传达室叫人的时候，只要按一下那个铃就可以了，那边的查皮就知道该起床了。G 大叔搬进去之后，会考虑到原有的查皮系统，留着叫醒查皮的那个开关。当用户启动电脑，G 大叔被叫醒的时候，G 大叔会一脸严肃的问用户：要用哪个系统？一个 ubuntu 一个查皮，给你 10 秒，快选！如果用户选我，G 大叔就来叫醒我，如果选查皮，G 大叔就去按那个铃。虽然 G 大叔说话有点不客气，不过还是尽职尽责的，作为一个启动管理器，就要负责好硬盘上每个系统的启动工作。可是查皮就不一样了，如果硬盘上已经住进了我，门房里已经有了 G 大叔，这时候重装查皮系统的话，查皮会不管三七二十一的把 G 大叔赶出来，在传达室装好他的“起床铃”就走了，不管我这边的情况。电脑再启动的时候，BIOS 自然就找不到 G 大叔了，就只能去按那个铃，直接启动了查皮，我的存在就完全被无视了。

那如果这样的惨剧不幸的事情发生了怎么半呢？他能把 G 大叔赶出来，我照样能在让 G 大叔再搬进去！想强拆？没门！不过，虽然说是这么说，这要想把 G 大树搬回去，前提是我得启动了才可以呀，处于睡觉状态的我是什么也干不了的。可是 G 大叔已经被赶走了，没法叫我起床了，我又怎么启动呢？还记得那张安装光盘么？还记得我说那是一张 LiveCD 么？还记得系统出问题的时候可以用他来修复么？没错，就是那张，赶紧让你家狗狗把他叼回来，现在用上了！用 LiveCD 启动电脑，就能够启动光盘上的 ubuntu 系统，这时候，电脑可就归我们 Linux 管啦！嘿嘿，小小的查皮算什么，你敢把我们的 G 大叔撵走？想的美！ LiveCD 启动之后，打开命令行，运行 sudo -i，获取权限。然后 mount /dev/sdax /media/ 这 sdax 就是你安装 Ubuntu 的时候用作根目录”/”的那个分区，如果你还单独分了 /boot 分区，那就还得 mount /dev/sday /media/boot/ 。当然，这里的 sdax,sday 都需要根据你的实际分区情况修改，可能是 sda1,sda4，或者 sdb2,sdc8，都没准。mount 好了之后，运行 grub-install –root-directory=/media/ /dev/sda 就好了。最后重启电脑，熟悉的 G 大叔又回来了。

当然，以上说的都是以后可能发生的情况，目前在我这里还没有这样的事情，隔壁那个查皮睡的死猪一样，不会有什么举动的。而 G 大叔早在安装的时候就自动设置好了多重系统启动，刚刚就是用户告诉 G 大叔来叫醒我去干活的。

起床之后，用户似乎对 G 大叔的举动不是很满意，打算要修改一下 G 大叔的配置文件。刚才我们说了，G 大叔启动的时候会去找 /boot/grub/grub.cfg 文件，这里面记录了一些 G 大叔应该做的事情。比如去哪里找我，去哪里叫醒查皮，等待 10 秒没有动静的话就默认叫醒我，等等。不过主人要想修改这些设置的话，可不需要修改这个文件，

而是要该 /etc/default/grub 文件。这个文件里，简单明了的记录了 G 大叔应该做的一些动作。只见用户下达了命令：sudo gedit /etc/default/grub 这命令的意思就是，以 root 用户的身份，命令 gedit 软件，去打开 /etc/default/grub 文件。之前我们说过，安装的时候创建的那个用户，不是一般的用户，是拥有能变身成 root 的能力的用户。这个 sudo，意思就是，我要变身！输入这个命令之后，我会要求用户再输入一边他自己的密码，注意，是当前用户的密码（比如我这里，就是 lanwoniu 这个用户），不是 root 的密码，真正的 root 用户的密码……是个迷。输入了密码，确认了他就是安装时的那个用户后，就可以以 root 的权限去执行后面的命令了： gedit /etc/default/grub 这个 gedit，是一个小的文本编辑器，要编辑文本文件，用他最方便了。gedit 小弟身轻如燕，迅速的从硬盘里爬起来，越进内存里，打开那个 grub 文件，显示在屏幕上。只见文件里写着：

```
#If you change this file, run 'update-grub' after wards to update
#/boot/grub/grub.cfg.

GRUB_DEFAULT=0
#GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release-i-s2>/dev/null||echoDebian`
GRUB_CMDLINE_LINUX_DEFAULT="quietsplash"
GRUB_CMDLINE_LINUX=""

#Uncomment to disable graphic alterminal(grub-pconly)
#GRUB_TERMINAL=console

#There solution use don graphic alterminal
#note that you canuse only mode swhich your graphic card supports via VBE
#you can see the minreal GRUB with the command` vbeinfo'
#GRUB_GFXMODE=640x480

#Uncomment if you don't want GRUB to pass"root=UUID=xxx"parameterto Linux
#GRUB_DISABLE_LINUX_UUID=true

#Uncomment to disable generation of recovery mode menuentrys
#GRUB_DISABLE_LINUX_RECOVERY="true"

#If you change this file, run 'update-grub' after wards to update
#/boot/grub/grub.cfg.
```

GRUB_DEFAULT=0 这行说的，就是让 G 大叔在用户没有选择的情况下，默认来叫醒我。因为在电脑启动的时候，G 大叔给用户的选项里，叫醒我，是排在第一个的。（但是 G 大叔数数喜欢从 0 开始数，所以是“=0”）用户似乎就是看着这个不大顺眼，把这个改成了 4。我掐指一算，改成 4，也就是启动时 G 大叔给用户的第 5 个选项……哦，是去叫醒查皮。用户的意思是，如果他没有选择，就让 G 大叔去叫醒查皮。哎……看来还是不喜欢我阿。另外这一行：GRUB_TIMEOUT=10 ，也被用户改成了 5。这行是 G 大叔等待用户选择的时间，原本是等 10 秒，如果用户不选择的话就去叫醒查皮，现在改成 5 秒了，看来这用户还是个急性子。改完了之后，保存了这个文件，gedit 小弟继续回硬盘里睡觉去了。但是还没完，光改了这个文件是不管用的，G 大叔真正关心的是 /boot/grub/grub.cfg 文件阿。还得运行一下 sudo update-grub，这样，就会根据刚才修改的 grub 文件，自动生成一个给 G 大叔看得，新的 grub.cfg 文件。这才算改完。

1.5 其他

从安装，到第一次启动，我还算比较顺利。可能是我遇到的这个用户运气好，也可能是我运气好，遇到个水平比较高的用户，到底是谁运气好，这是个哲学问题……不过总之，会有一些人安装 Ubuntu 不是那么顺利，或者因为种种原因安装遇到阻碍，不过没关系，没什么困难是不能克服的。

有人说了，我就遇到困难了。你刚才讲了这么半天，那么多步骤，还得分区，还得设什么 BIOS，太复杂了，听不懂阿。有简单点的办法没有？我告诉你，有。

为了更多已经装了查皮之类系统的人能够更加简单的安装和体验我们 ubuntu 系统，我们的光盘里带了一个软件，叫做 wubi。您可听明白了，他可不是个输入法，不要妄想用五笔字型输入“我要装系统！”就能把 Ubuntu 装上。这个 wubi 是 WindowsUBuntuInstaller 的缩写，这家伙是运行在查皮系统下的软件，他的功能，就是帮助你在查皮系统下安装我们 Ubuntu 系统。不用你懂分区，不用你知道挂载，不用改变当前硬盘的状态，一切全都交给他就好。并且我们这个光盘还设置好了自动运行，光盘放进去就会看到 wubi 运行的界面了，就这样：

第一个选项，就是之前说的光盘安装。选了这个选项之后电脑就会重启，然后从光盘启动（还是得设 BIOS），之后就跟我们说的光盘安装没有区别了，不选这个。（不选你说那么半天！）看第二个选项，“在 windows 中安装”，这个看着新鲜吧？好，就是他了！

点击之后，就看到一个设置的页面：

目标驱动器，就是让你选择把 ubuntu 装在哪个盘上。装的时候会在你选择的哪个盘上创建一个巨大的文件，文件的大小就是下面那个“安装大小”里选的大小，这个文件就会被 Ubuntu 当作硬盘来用，所以可以尽量大一点。既然是把文件当硬盘用了，自然也就省去了调整分区之类的麻烦了，怎么样，很人性吧。另外，由于 FAT32 支持的文件大小有限，所以你选择的目标驱动器需要是一个 NTFS 格式的。

安装大小，不多说了，就是用来当硬盘的那个文件的大小，也就相当于让你选择划分多少空间给 Ubuntu。这个一旦确定，装好 Ubuntu 系统之后，可就不好该了，所以一定要想好。如果想做日常使用的话，至少要 20G，如果只是装来看看，10G 就够了。

桌面环境，没啥说的，就是 Ubuntu。

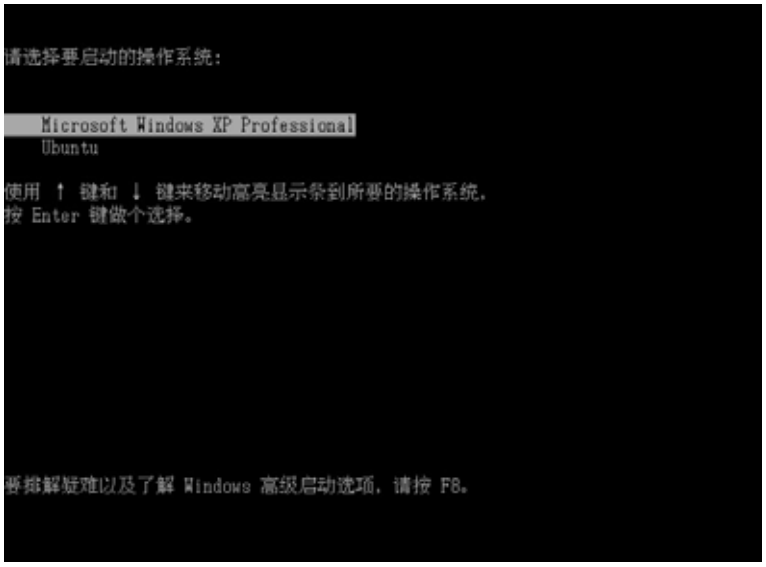
语言，你说呢？

用户名，就是安装时候创建的那个有变身能力的用户。密码，你知道的，国际惯例。

都选好了，自然就是点“安装”

之后就是毫无悬念的进度条，这个过程结束之后会问你要不要重启。如果你正跟小妹妹聊的火热，待会再重启也不妨，如果没什么事情，那就赶快重启吧。

重启之后会看到系统选择的界面，如果装过多个 windows 的话会很熟悉，就像这样：



选哪个？还用问么，自然是 Ubuntu。选择之后就进入安装的第二个阶段，第二阶段也只是没有悬念的进度条而已，等着就好了，装好了再重启，OK，可以进入 Ubuntu 了。

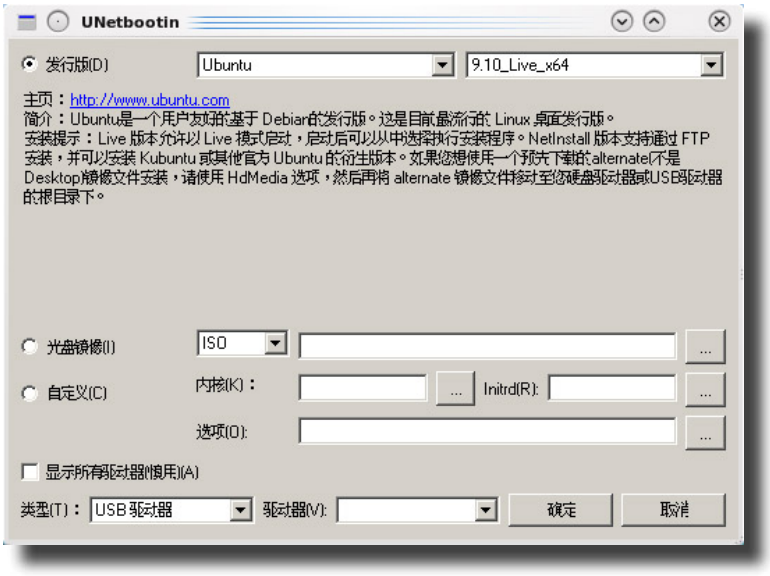
又有人说了，我这还有困难。你上面说的方法都得用光盘阿，可我的电脑没有光驱，这可就没法装了吧。不管是申请来的光盘还是自己把 iso 刻录成光盘都得有光驱呀。别急，光盘地没有，U 盘地有不？有？那也有办法！

首先呢，我给您先算上一挂。我之前说过，要想找我们 Ubuntu 到你的电脑上工作来，要么就从网上下载，要么就申请光盘。既然你没光驱，那大概你不会去申请光盘。所以你是从网上下载的 ubuntu 的 iso 文件，对不对？并且鉴于那个稍微有点软公司和查皮的知名度，你的电脑里有 60% 的可能性已经装了查皮这个系统并且在这个系统下下载的 Ubuntu 的 iso。另外 40％的可能性是装了是查皮的后代系统，什么“温妻”阿，“喂死它”之类的。又猜对了吧？好，如果我都猜对了，您也不用鼓掌，接着往下看。如果我猜错了，那就是说您竟然在没有光驱的情况下申请了光盘并且还想装上，或者您在电脑上没有系统的情况下，神奇般的下载了 ubuntu 的 iso 的话，那这段就别看了，看了也没用。

废话不多说（这还少么……），咱们开始正题。咱拿查皮那个系统来说事，其他的什么“温妻”，“喂死它”也是一样的道理。在查皮系统下，有这么个软件，叫做 UltraISO，这家伙本事挺大，可以用来刻录光盘。“我没光驱阿，刻录的哪门子光盘阿！”您别着急，我还没说完呢，他不但可以刻录光盘，还能刻录 U 盘。把您的 U 盘准备好，里面的内容赶紧现找别的地方存起来，一会就啥都没了。而且记得要用 U 盘哦，别拿个 mp3 来糊弄，挂了我不负责。把 U 盘插在电脑上，运行 UltraISO 软件，打开 ubuntu 的 iso 文件。然后选择“启动 --> 写入硬盘映像”，在弹出来的对话框中的“硬盘驱动

器”选项选择好你的 U 盘，可一定要选对哦！否则就指不定丢多少数据了。“写入方式”选择 USB-HDD+，然后，就开始吧！等写入完了，你的启动 U 盘就作好了，用他启动电脑就像用 liveCD 启动电脑一样了。不过要说明的一点，这个 UltraISO 可是要付费的，别偷来就用。“我平时也不用，就为了刻录一下还得付费？有点亏阿”恩，可能是有点，如果您不想付费的话，没关系，咱还有办法。

还有个软件，叫 unetbootin。这个家伙可是个免费的开源软件，可以到这个地方来找他：http://unetbootin.sourceforge.net/ 这软件同时支持查皮系统和我们 Linux 系统，您既然是想在查皮下创建 ubuntu 的 liveUSB，那自然要下载那个查皮版本的。从网上把这个软件下载到你的机器上，并运行之后，就出现软件的界面：



你已经下载了 ubuntu 的 iso 文件，所以选择“光盘镜像”，然后选择存在硬盘上的 ISO 文件，再选好驱动器就好了。（注意，驱动器一定要选择对！否则的话指不定你哪个硬盘的数据就挂了！）确认都选对了之后，点“确定”，就等着好啦。

另外，去过我们学校下载 Ubuntu 的同学也许发现，那里除了 ubuntu-10.04-desktop-i386.iso 这个 LiveCD 以外，还有很多其他版本的 ubuntu，比如什么 ubuntu-10.04-desktop-amd64.iso,ubuntu-10.04-dvd-i386.iso,ubuntu-10.04-dvd-amd64.iso，ubuntu-10.04-alternte-i386.iso 等等。这么多种，有什么不一样呢？等我慢慢说给您听。

先说这个，ubuntu-10.04-desktop-amd64.iso。这个和 ubuntu-10.04-desktop-i386.iso 唯一的区别就是一个是 i386，一个是 amd64（废话，傻子都知道！）这 i386 和 amd64

说的是 cpu 类型。有同学说了：“哦，那我知道了，这个 i386，因为有 i 嘛，就是用在 Intel 公司的 CPU 上的，那个 amd64 自然就是用在 amd 公司的 cpu 上的。”我很高兴的告诉这位同学：“你答错了！”i386 指的是 x86 架构的 32 位 CPU, 因为这种架构是在当年人特二公司生产 Intel386 处理器时就确定下来的，所以叫做 i386. 之后的什么奔腾几，奔腾几的，都是这个架构。还有个阿妹哒公司 (AMD)，也生产兼容 x86 架构的 CPU，一大堆这个龙那个龙的，都是 i386 兼容的 CPU。后来随着技术的发展，32 位的 CPU 逐渐退出了历史舞台，出现了 64 位的 CPU，至于具体什么是 32 位，什么是 64 位，他们有什么不同，咱们以后会详细说。最先推出 64 位 CPU 的，就是那个阿妹哒公司，所以目前 64 位 CPU 的这种架构是阿妹哒公司确定的，于是就叫做 amd64。那什么 CPU 才是 64 位 CPU 呢？基本上你现在能买到的全是！（在潘家园买到的不算阿）刚才那位同学又说了：“哦，那我明白了。i386 就只能装在 32 位的古董级 cpu 上，amd64 就只能装在主流的 64 位 cpu 上。”我再次恭喜这位同学——又错了！考虑猜到现在的很多软件依然不支持 64 位，所以现在的家用 64 位 CPU 都是兼容 32 位的，也就是说在 64 位 cpu 上安装 i386 的系统是可以的，比如我所在的这台电脑就是 64 位的，可是隔壁的查皮就是 32 位的系统。但是要在 32 位 cpu 上安装 amd64 的系统，那确实不行，想都甭想。

然后咱再说说这个 ubuntu-10.04-dvd-i386.iso。最后的这个 i386 不用解释了，跟上面一样。主要就是这个 dvd。其实这个跟 desktop 的区别就是，那个是 CD 的，这个是 DVD 的。（又跟没说一样……）这个 DVD 的里面比那个 CD 的多了一些常用的软件和语言包。不过有一点，默认安装的软件和 CD 版的是一样的，别以为 DVD 的就多给你装什么软件。只不过装完系统之后，可以从光盘安装其他的软件而不用上网去下载了。这主要是针对上网不大方便的人的，有一张 DVD，基本软件就都齐了。另外还有一点好处就是，如果你选则的语言是汉语的华，DVD 版的装好了之后有比较完整的汉化，因为 DVD 容量大，可以装下更多的语言包。

至于这个 ubuntu-10.04-alternte-i386.iso。alternte 的意思就是安装的时候可以选择安装的软件，可以装成桌面版，也可以装成服务器版，可以有图形界面也可以没有图形界面，总之，有很多选择，是给高手们预备的。另外安装的时候是没有图形界面的，需要英语好，对 Linux 系统熟悉的同学才可以装。行了，关于安装就说这么多，我要干活去了，主人要上网呢。

Chapter 2

第二章

渐入佳境

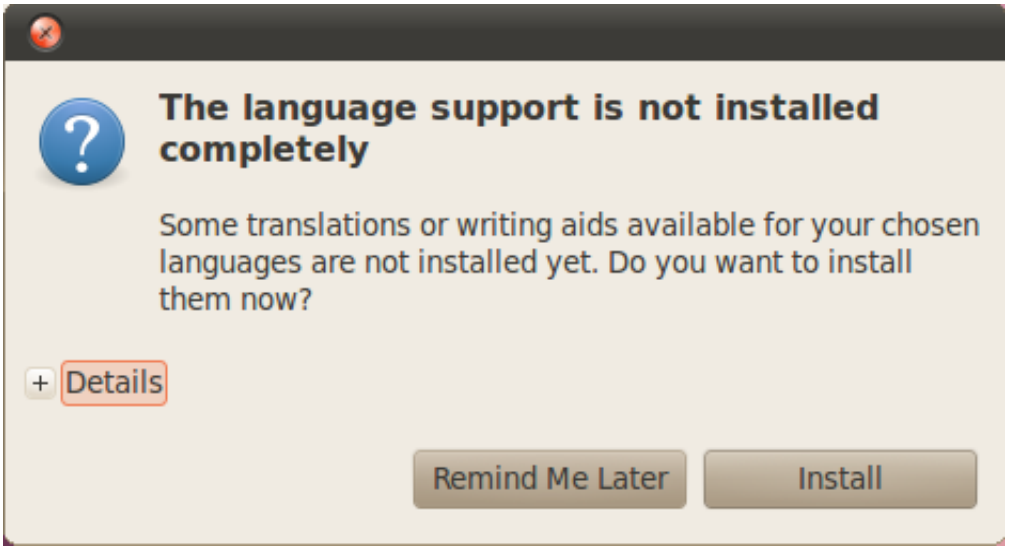
2.1 超级牛力

用户修改了启动顺序后，又重启了一下，大概是想看看修改后的效果。于是，BIOS 再一次被电醒，G 大叔再一次面无表情的问：要叫哪个？给你 10 秒，不说我就去叫醒查皮了阿！ 用户微笑的点点头，看来修改是有效果的。他没有进入查皮系统，还是选择了我，结果，刚刚躺下还没把被子捂暖的我，又被 G 大叔叫起来干活了。

用户先点开了屏幕上方的 Applications 菜单，接着鼠标又移动到 Places 菜单，然后是 System 菜单，最后鼠标停了下来。我估计，他是在思考，思考着一个关于他记忆力的问题：“我装的时候选的是中文吧，怎么全是英语呢……”

呵呵，关于这个问题，主要是由于我汉语课没好好上，逃了那么 7、8 节，谁还没逃过课呢，你说是不？所以呢，也就记不住那么些中文。按说记不住没关系，我随身带本字典也就是了，不过由于我是乘坐 CD 光盘来的，那上面的空间实在有限，装不下那么厚一本汉语字典了。所以我只好凭我仅有的一点记忆，翻译了几个简单的词，于是用户也就看到只有少数几个地方被翻译了过来，大部分的菜单还是英文。

但是用户不知道这其中的缘由阿，于是他最终决定要为自己的记忆力讨个说法，他点击了 System-->Administration-->Language Support 菜单，想要看看设置的到底是不是中文。点开之后，我很诚实的告诉他：你选的确实是中文，但是呢，由于一些错综复杂的原因（比如我逃了汉语课），相关的一些语言包并没有完全安装，那么你现在要不要马上装呢？



用户义无反顾的点了”Install”，于是我赶紧去硬盘里叫醒一个人——apt。他是谁呢？

要说 apt，先得说说我们 Ubuntu 安装软件的方式。您可能用过查皮那个系统，那系统上装软件怎么装呢？先上网，搜索你要装的软件，从 n 个网页中找到一个没过期的，点开，再从网页上数十个硕大而带有欺骗性的广告的缝隙中找到真正的下载连接列表，根据你是电信还是联通还是什么网的，选择合适的连接，下载，下完了后双击，安装，选择装到哪，下一步，下一步……完成。大概就是这样是吧。我们可不这么装，太麻烦！我们 Ubuntu，以及大多数的 Linux 都有自己的软件包管理器，要装什么，只要跟这个软件包管理器说：我要装 xx 软件，然后等着就好了。不用你自己上网找，不用你选择装在哪里，软件包管理器会负责全部的工作。apt，就是我这里的软件包管理器。

这个家伙的角色就像个公司里的人事部经理兼后勤部长，装个软件卸个软件的，都是他管，软件所需要用的各种环境文件啥的，也都是他负责准备。当别人夸奖他的时候，他总是自信的拍拍自己的胸脯说：“本 APT 有着超级牛力”。而他也确实很厉害，很敬业，也很专业，对于人才（对我来说也就是软件）的各种情况了如指掌。要招一个人来的时候，他会做好所有准备工作，这个人需要用什么样的库，或者需要什么其他的人才能一起协同工作，他都会事先做好准备。比如，用户想用 vim 来编辑文件，就叫 apt 去招 vim 来。apt 就会报告，说 vim 要来的话，首先需要准备好 libncurses 这个库，和 python 这种脚本语言的执行环境。征得同以后，他就会去网上找这些东西，并且运回家，把库放在该放的地方，相关的软件安排好住宿，然后再去找 vim 同志，请他过来帮忙干活，并且说明，环境都已经布置好了。每次新人来了之后都很感谢 apt 同志为自己做的这些准备工作，该有的东西，该来的助手都在，于是干活就事半功倍了。但把人才请来之后，apt 同志的工作还没有结束，他还要把现在的人事情况记录下来，以便哪天用户问起来：“我说超级牛力阿，咱这现在都有多少软件阿，都是谁阿？”apt

好能从容的回答。可以说，apt 这家伙对于我来说实在是非常重要的，有了他，Ubuntu 才是 Ubuntu。所以，在我们这里，要忘记查皮那种安装软件的方式，装软件就跟”超级牛力”说，让他去办去，省心。

这不，用户要装中文环境，我就叫来了超级牛力，给他列个清单，包括中文的字体阿，输入法阿等等相关的软件和资料什么的，一大堆，告诉他，把这些给装上。超级牛力拍拍胸脯说：放心吧，本 APT 有超级牛力！说完一转身，跑出去，到网上下东西去了。要说超级牛力的工作热情我是了解的，干活向来很麻利。可是今天我在内存里等了足足有一秒钟，才见超级牛力拖回来 1k 的数据（也就是 1024 个字节）。我赶紧拉住他问：“怎么回事？怎么这么慢阿！照这速度得什么时候下完了阿。”

他叹了口气：“唉～别提了，这路也太远了，得翻过 6 个路由，跨过 8 道防火墙，路还窄，不是车多流两大就是行使缓慢。费好半天劲才拖回这么一点，行了，不跟你多说了，我赶快再去拖去，本 APT 有超级牛力～”之后喊着口号又走了。留在内存里的我只好如实报告给用户：这个……目前的下载速度大约每秒钟 1K 左右，预计能在今年端午节前下完……1 秒后，超级牛力又进来了，这回拖回来了 1.1k 数据，比上回多点，可也有限。我忍不住问他：你是去哪下的这些东西阿？

超级牛力说：“就是那 http://cn.archive.ubuntu.com/ubuntu/，本 APT 有超级牛力～～～”唉～这家伙，我知道是怎么回事了。

这要从超级牛力的工作原理说起。

超级牛力会自动上网去下载软件，但他可不是四处瞎找，而是去固定的地方找。这个固定的地方有个名字，叫做软件源。那里专门为每个 Ubuntu 上的超级牛力提供各种打包好的软件以及相关的信息介绍，供超级牛力们下载。这样的软件源有很多，遍布世界各地，超级牛力应该去哪个呢？其实他自己也不知道，他需要一个列表，一个软件源列表。这个列表就写在 /etc/apt/source.list 这个文件里，这里面记录着软件源的地址。超级牛力工作的时候，就会去这个文件里记录的地方去下载软件。那现在系统刚刚装上，那个文件里默认的软件源地址不可能适应所有的人，是不是？有的人可能离的挺近，去默认的源就很方便，有的人去默认的源可能就很费劲，没准得绕过半个地球才行。默认的源不够快，自然就需要找个快一点的源来代替。

现在我这位用户就是因为没有设置好软件源，结果就不得不考虑：究竟是等中文包下完了再去买粽子叶，还是等吃完粽子再去感受中文的 Ubuntu。终于，他作出了明知的决定——取消！换源！

去哪找软件源呢？可以去我们学校打听，也可以上网搜索，或者去论坛问问，途径很多。我这位用户发话：上网！于是我赶紧叫醒 Firefox 起床干活，她是默认的浏览器，我喜欢叫她狐狸妹妹。狐狸妹妹轻移玉步，走进了工作间——速度有点慢，不过还可

以接受，然后开始工作。用户让去那个啥狗狗哥搜索网站，搜索了一下”ubuntu 软件源”，找到

```
deb http://mirrors.163.com/ubuntu/ lucid main universe restricted multiverse
deb-src http://mirrors.163.com/ubuntu/ lucid main universe restricted multiverse
deb http://mirrors.163.com/ubuntu/ lucid-security universe main multiverse restricted
deb-src http://mirrors.163.com/ubuntu/ lucid-security universe main multiverse restricted
deb http://mirrors.163.com/ubuntu/ lucid-updates universe main multiverse restricted
deb http://mirrors.163.com/ubuntu/ lucid-proposed universe main multiverse restricted
deb-src http://mirrors.163.com/ubuntu/ lucid-proposed universe main multiverse restricted
deb http://mirrors.163.com/ubuntu/ lucid-backports universe main multiverse restricted
deb-src http://mirrors.163.com/ubuntu/ lucid-backports universe main multiverse restricted
deb-src http://mirrors.163.com/ubuntu/ lucid-updates universe main multiverse restricted
```

要换源很简单，把 /etc/apt/source.list 文件打开，把里面东西清空，换上上面这一大坨就好了。不过我也知道，看上去容易的事情，到了新手那里，都会出问题。

我这位用户找到了一个看上去还不错的源，也知道要改哪个文件，于是，他动手了。他对命令似乎还挺熟悉，只见他运行了：gedit /etc/apt/source.list 。

这个命令的意思是，叫 gedit 出来，打开 /etc/apt/source.list 文件。于是我赶快去硬盘里叫醒 gedit 小弟。gedit 是一个文本编辑器，比查皮那里的记事本稍微强大点，同时身体灵活，动作敏捷。瞬间，gedit 小弟就来到了内存里，比狐狸妹妹可快多了。gedit 赶快打开那个文件，显示给用户。用户把里面的东西统统一删，把找到的软件源的地址粘贴了进去。然后，很自然的想去点“保存”。咦？怎么竟然点不了呢？ gedit 在那冷嘲热讽的念叨：“你以为是谁呀你？这文件可是重要的系统文件，给你看看就得了，你还想改？！还想存？！改坏了算谁的呀，你一个普通用户你担待的起么你。”当然，这些都是他在工作间里自言自语，并没有真的把这些话显示给用户，否则就等着被删除吧。那到底为什么用户不能保存呢？原因很简单——没有权限。

“不对呀，你不是说安装系统的时候建的这个用户有超级用户的权力么？”不好意思，您少记了仨字，我说的是，这个用户有“变身成”超级用户的权力。怎么变身？扭回头来吧，不用看窗外的月亮，只需要在命令前面加上 sudo。sudo，就是“以超级用户的身份运行”的意思。用户直接运行 gedit /etc/apt/source.list，就是以普通用户身份打开 source.list 文件，当然不能修改。应该运行

```
sudo gedit /etc/apt/source.list
```

就是用户明确的说：我要变身啦！我要以超级用户的身份打开 xxxx 文件。但是不能您说变身我就让你变，还得讲讲条件。首先是核对一下身份，只有最初安装系统时候创建的那个用户可以变身，装好之后再创建的用户就不行了。（当然，最初的那个用户也可以把变身的能力赋予其他用户，这里说的只是默认情况。）确认了这个用户就是最初安装系统的那个用户之后，还不算完，还需要让用户再输入一遍自己的密码。这一来是防止恶意程序脚本骗取超级用户权限，再者也可以确认现在坐在电脑前的就是登录进来的用户。别回头老陈登录进来了，正处理着半截照片，上厕所去了，这时候来个修电脑的偷偷在老陈的电脑上以超级用户身份搞破坏，那就容易出事了。再有一点，输入密码的时候是不会回显的，不像图形界面输密码的时候会显示出相应个数的圆点。在命令行里输密码，什么星号阿，圆点阿，统统的没有，你就输就行了，千万不要怀疑自己的键盘过保修期了。

本以为我这位用户会困惑好一会为什么不能保存，结果发现他好像不是不知道 sudo，只是一时忘记了，发现 gedit 不能保存之后，马上就把 gedit 关了，在命令前加上了 sudo 重新来了一遍，这回 OK 了。看来这家伙还是个老手嘛，只是一时忘了。我开始庆幸我能遇上这么一个用户了。

软件源修改了之后，还不能马上生效，得先通知超级牛力一声，让他去根据软件源更新软件列表。这列表是怎么个意思？这个列表就是写明，现在所用的这个源里面都有些什么软件，都有什么依赖关系，这样当你要装软件的时候，超级牛力直接查看这个列表就知道这个软件有没有，这个软件都依赖什么其他的软件等等信息了。否则的话，你告诉超级牛力，要装 give_me_money 软件，超级牛力还得大老远跑到网络上，找到软件源服务器，问人家：您这有一个叫 give_me_money 的软件么？人家说：我还想要呢！没有，回去！然后超级牛力再回来告诉你。这样很耽误时间。所以，就要在每次换源之后，让超级牛力去那个新的源问好了，都有什么软件，每个软件都什么版本之类的信息，把这些信息存在硬盘上。你要是再想装什么不靠谱的软件，就可以直接让你死了这条心，不用跑到网络上问了。那怎么更新呢？简单，一个命令

```
sudo apt-get updata
```

软件源也改了，列表也更新了。这回，用户再打开 System-->Administration->Language Support 菜单，提示安装的时候再点 install——这回，只需要吃一个粽子的时间就可以下载完了。

当然，刚才所说的这一切的前提是，你要把网络配置好，能上网，才能发挥超级牛力的能力。我所在的这台电脑因为是用那种家用的宽带路由器，什么 IP 地址阿，

DNS 阿，路由阿等等，都是由路由器自动分配的，所以在我这里不需要设置什么，网线插上就能上网了。如果不是这样的怎么办呢？

比如有的是需要自己手动设置网络参数，也就是 IP 地址阿，DNS 阿之类的。这个好办，你在查皮下怎么设置的来着？找网络连接是不是？那我这里还找这个就对了，就在系统 --> 首选项 --> 网络连接。第一个标签就是 " 有线 "，下面列出了所有的有线网卡，需要用哪个，选中，然后点右面的 " 编辑 " 就好了。点开之后就可以设置 MAC 地址，IPV4 的地址（也就是我们平时看到的 xxx.xxx.xxx.xxx 形式的地址。），甚至 IPV6 的地址。IPV6 用的还不广泛，咱就光说这 IPV4 吧。在 IPV4 标签里，“方法”一栏，原来可能是自动，也就是依靠 dhcp 分配地址，要手动指定就选手动。然后在下面点添加，然后写上 IP 地址，子网掩码，网关。下面再填上 DNS, 就好了。你问我具体应该填什么？跟查皮下一样！

还有的，是需要拨号的，比较常见的就是 ADSL，这个也好办。还是网络连接，最后一个标签，就是 DSL。切换到 DSL 标签，点添加，写好用户名密码（当然是 ADSL 的用户名密码了。） 设置好 IPV4 的参数（一般用自动就好了），就可以了。

如果是无线局域网，那就是无线哪个标签，如果是宽带上网卡，那就是移动宽带哪个标签，总之，大多数的上网方式，都是可以的。好，说了这么些，这会那个中文支持包也安装完了。

2.2 狐狸妹妹

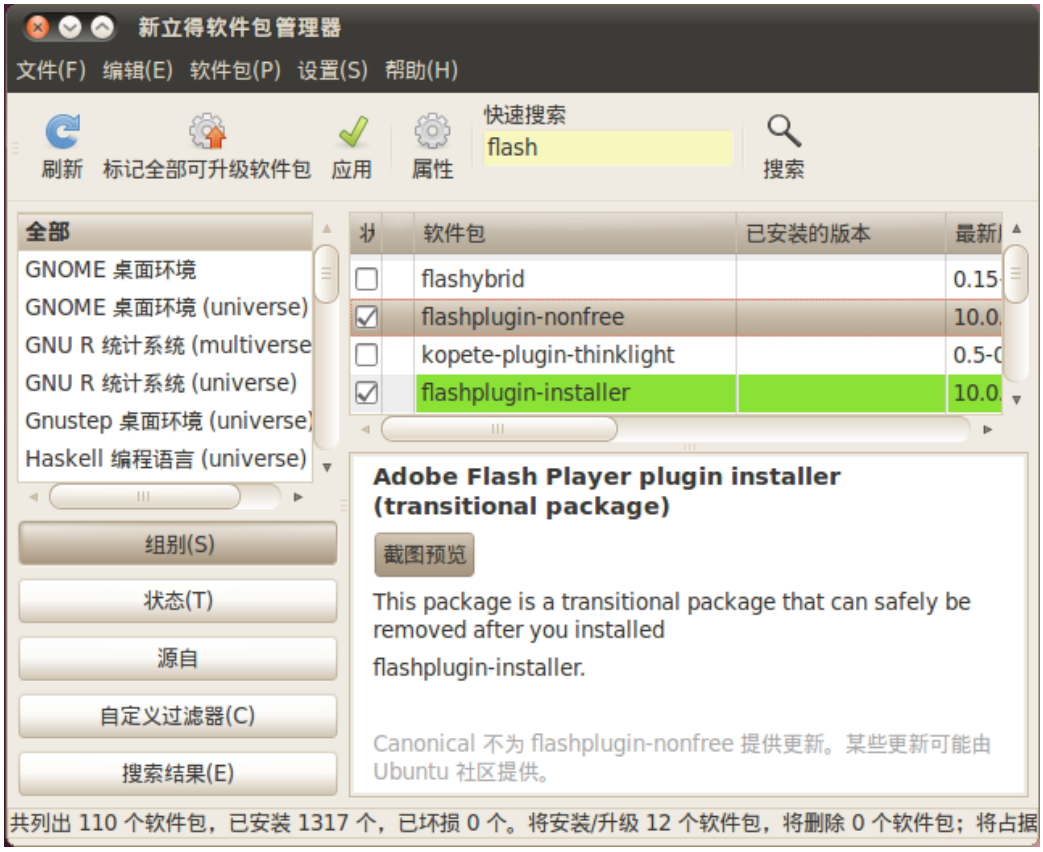
过了一个粽子的时间，中文支持装好了。主人注销并重新登录了一下，总算看到了熟悉的文字。（主人，就是我那用户啦，看得出来他是要把我当作日常主要的系统来用，所以我已经认他做我的主人啦。）不过，这只是万里长征走完了第一步。从开始安装到配置好中文，已经过了很长时间，主人喝口水，喘口气，忽然想起一件很重要的事情——菜该收了！于是他再次叫醒狐狸妹妹，出发了。

主人指引着狐狸妹妹来到了他的菜地，可是放眼望去，怎么什么也没有，一片空白阿？就算这菜被人偷光了，难道这地也让开发商强占了不成？再仔细抬眼一看，哦，原来狐狸妹妹已经做了解释：您需要其他插件以显示此页面的所有媒体。

其实很简单，就是没装 Flash 插件而已，那就装上吧。主人点了狐狸妹妹提供的那个“安装缺失插件 (I)”的按钮，之后狐狸妹妹就跑到网上去找插件。狐狸妹妹装插件跟超级牛力装软件有点类似，都不需要您手动去下载，都是他们自己去网上找来装。但是狐狸妹妹的插件只在指定的官网上有，只此一家，别无分号，所以也不用专门为狐狸妹妹设置什么软件源。废话不多说，之间狐狸妹妹上网上搜索了一遍之后向主人报告：我搜了，可是呢……没找到，您自己看着办吧。这要是换了别人，早气得把她关了，好在我主人脾气还比较好，并且可以理性的分析问题。狐狸找不到 Flash 插件没关系，咱再找别人嘛。不就是装个 Flash 插件么，flash 插件也是个软件，既然是装软件，那就找超级牛力就没错，人家可是专业的。于是，主人点击了：系统 --> 系统管理 --> 新立得软件包管理器。

等等，不是说要找超级牛力么，这个新立得又是个什么东西？

超级牛力确实很牛力，干活没的说，但他是个命令行界面的软件，只能够通过文字跟用户交流。可是由于人类越来越懒，键盘是能少敲一下就少敲一下，因此很多人并不喜欢跟字符界面的软件打交道，这时候，就该新立得出场了。新立得就是超级牛力的图形界面前端，用户通过喜欢的图形界面，把自己的意图告诉新立得，再由新立得转达给超级牛力。这两个家伙配合的很是默契，以至于很多人觉得，新立得就是超级牛力。主人点开了新立得，由于新立得是用来给系统安装软件的，需要变身成超级用户才可以操作，因此新立得马上要求主人输入他的密码，核对身份，核对通过之后，才显示出真正的，新立得的界面。



在新立得的界面上装软件就像逛超市一样，所有可以装的软件都给你列出来了，左面还分好了各种类别，想要什么，勾上就行了。有时候会有一些依赖关系，所谓依赖关系，就是要装软件 A，必须先装软件 B；要买牙膏，就必须先得买牙刷（否则，总不能用鞋刷子刷吧。当然，如果你已经买了牙刷那就另说了）。新立得比超市先进的地方，就是能搜索。主人要装 flash，于是就在 " 快速搜索 " 栏里面输入 flash，找到很多跟 flash 有关的软件包，其中有一项 "flashplugin-nonfree"。主人看了一下对这个软件包的介绍，觉得没错，就是这个，然后右键点击这个软件包，选择了：标记以便安装。新立得赶紧叫起来超级牛力说：“牛哥牛哥，快起床，主人要装 flashplugin-nonfree 这

个包啦。”超级牛力立刻投入工作，查阅了一下自己记录的资料，告诉新立得：“转告他，要装 flashplugin-nonfree，就得同时装上 flashplugin-installer、ia32-libs、lib32asound2、lib32bz2-1.0、lib32gcc1、lib32ncurses5、ib32stdc++6、lib32v4l-0、ib32z1、libc6-i386nspluginwrapper 这些软件包”新立得如实转告，主人倒是没有被这么多乱七八糟的包名吓倒，很淡定的答应了新立得的要求。于是就开始安装了么？没有，都说了新立得装软件就跟推着推车在超市购物一样，先选择所有你需要的，最后一起算钱。不过这次主人只需要 flash，于是点击了新立得界面上的“应用”按钮，这按钮的意思就是：结帐。

经过一段时间的等待，超级牛力装好了 Flash 插件，并由新立得汇报给了主人。主人迫不及待的牵着狐狸妹妹再次奔向了他的菜园。　　这回一眼望去，果然绿油油一片，长势喜人。咦？菜地里怎么这么多麻将牌呢？再一想，不对，哪有一地麻将牌的道理，仔细一看，哦，原来是所有中文都变成方块了。主人心里肯定嘀咕：我不是都装了中文了么，系统里的其他地方都变成中文了，怎么这里还是方块呢？百思不得其解的主人让狐狸妹妹赶快去狗狗哥那网站去问问。狐狸妹妹干活利索，很快就问回来了，原来，只是 Flash 默认使用的字体不对，只要修改配置文件，换个字体就好了。

于是主人按照网上找到的方法，打开了 /etc/fonts/conf.d/49-sansserif.conf 文件，这回他记得要加 sudo 了，因为是 /etc/ 下的文件嘛，肯定只有 root 才有权限的。然后把里面写明使用 sans-serif 和 serif 字体的地方，都换成了 wqy-zenhei 字体，最后保存。改完了就类似这样：

那么换上的这个 wqy-zenhei 是个什么字体呢？这个就是大名鼎鼎的文泉驿正黑。

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
<!--
 If the font still has no generic name, add sans-serif
 -->
    <match target="pattern">
        <test qual="all" name="family" compare="not_eq">
            <string>wqy-zenhei</string>
        </test>
        <test qual="all" name="family" compare="not_eq">
            <string>wqy-zenhei</string>
        </test>
        <test qual="all" name="family" compare="not_eq">
```

```
                <string>monospace</string>
        </test>
        <edit name="family" mode="append_last">
                <string>wqy-zenhei</string>
        </edit>
    </match>
</fontconfig>
```

各位要有兴趣，我说说，您听听，在想当初，很久很久以前（对你们人类来说其实也不算久啦，也就 6，7 年以前），那时候的 Linux 日趋完善，不少国内的开源同好们，都来尝试安装各种各样的 Linux。虽然硬件兼容的越来越多，应用程序的安装越来越便利，但中华儿女们安装了 Linux 之后无一例外的遇到了中文化的问题——没有一个合适的中文字体。要知道，Linux 是自由的，开源的，其中很多是免费的。那么自然不可能在免费的 Linux 中带有任何需要付费的字体。那 Linux 上自带的免费字体是从哪来的呢？都是世界各地热心的爱好者们贡献的。爱好者们自己创作一套字体，然后无偿贡献出来，给大家使用。因此，有很多优秀的，免费的英文字体可以用。可为什么其他国家有热心爱好者贡献字体，就没有热心的中国人贡献字体呢？是因为中国人懒？不是。是因为中国人自私？也不是。是因为中国人少？靠，更不是了。答案很简单，因为中国字多！人家做一套英文字体，总共也就二十六个字母的大写加小写，外带十个数字和一些标点符号，加在一起不超过一百个。一个人花一周时间就能做完了。汉字有多少？找本新华字典翻开前言看看——收录汉字一万个左右！这要是让一个人把这一万个汉字都做成电脑中的字体，还不得累吐了血阿。就算是只作常用汉字也得有四千多个，这还不算繁体字和各种数字，标点。那么那时候有没有中文字体呢？当然是有的，否则难道十年以前中国人都不用电脑？可是一般中文字体都是需要收费使用的——这个很合理吧，这么困难的东西，人家一个人（也可能几个人）费了好大劲作出来了，人家也得穿衣吃饭，养家糊口嘛。就算有几个不免费的中文字体，也是有很多问题，丢字阿，显示效果不好阿之类的。所以那个时候，Linux 的中文用户就只有忍受着质量差，总丢字的中文字体，或者偷偷的把其他系统下的付费字体复制过来用。说起来偷付费字体这件事，虽然不至于今天偷来用了，明天警察就来敲你家门，可毕竟不算光明正大不是。

直到 2004 年，中文字体的事情有了转机，带来转机的，是一位美国哈佛大学医学院的博士。这位仁兄倒不是来发扬国际主义精神的，他之所以关注中文字体，是因为他本身是个中国人——房骞骞博士。不知道是不是因为他家房子被拆迁了，所以被逼无奈去美国当博士去了，反正他在接触到 Linux 的时候，发现缺少中文字体是个很头疼的事情。他也知道要是自己作出整套中文字体一定要吐血的，但是他还知道，积少

成多，集腋成裘，众人拾柴火焰高，一个好汉三个帮，一个篱笆三个桩……（此处略去类似俗语若干）他利用网络为平台，创建了“文泉驿”项目，目的是要创作出一套高质量的，免费的中文字体。他自己动手编码，设计网站，简化设计字体的复杂程度，把汉字字体的设计搬到了网页上。这让每一个热心的志愿者都可以按照网站上的指导，完成一个个汉字字体的设计。就这样借助全球草根志愿者的力量，他开始了“万里长征”。经过数年的连续奋战，至今取得一项永留史册的硕果。这就是“文泉驿”汉字库，其中包括点阵和矢量字体。文泉驿正黑，就是其中之一。

主人改好了配置文件，看看菜地，还是一地麻将牌。忽然想起什么，把狐狸妹妹关掉重新启动了一下，恩……这回好了，终于看见Flash里的中文了，Flash算是配置好了。Flash 技术在现在的网页中用的是相当广泛，很多在线的视频阿，音乐阿，游戏阿，都是 Flash 的，把 Flash 弄好之后，主人再用狐狸妹妹上网，就基本没有什么障碍了。主人很高兴的想把这个好消息告诉给他的好友，忽然发现，咦？还没有聊天软件呢。

2.3 心有灵犀

这么重要的软件怎么会没有呢？主人也马上意识到了这一点，于是赶快点开了应用程序 --> 互联网，一看，有个“Empathy 即时通讯客户端”，恩，估计就是他了。

Empathy 这个家伙其实就是个送信的，如果你查字典的话，会发现 Empathy 这个词的解释是：感情移入，同感（对他人的感情，经历等的想象力和感受力。） 解释的比较复杂，于是我们就叫他 " 心有灵犀 " 吧。心有灵犀这家伙虽然送信，但他不管送那种长篇大论的 Email，而是负责发短消息——short message，也有叫短信的。不过别误会，这可不是说手机的短信，而是指像 msn 啊，qq 啊这样的即时通讯的消息。这些聊天软件的工作都是送信，使用者把要说的话写成信给他们，他们把信放在信封里——这个过程叫打包，然后把这个包发送给对方的软件。对方软件拿来这个包，先要拆包——也就是吧信从信封里拿出来，然后把里面的内容显示给用户看。可是这些软件互相之间是不能通信的，msn 不能给 QQ 发消息，反过来也不可能。这个估计不用我多说，用过查皮下的这样软件的都知道。这是因为他们的信封——打包方式不一样。比如 msn 的信封可能是从上面拆开取信；QQ 的信封则是从侧面拆开取信；Gtalk 的信封可能是用订书器订上的，需要拆钉取信；而百度 Hi 的是用胶水粘上的，需要涂水溶胶取信。反正各有各的高招。那么心有灵犀呢？他全会！否则对不起他的名字阿。

心有灵犀在学校的时候，专门学习过各种各样的通讯工具信封的封、拆方法。什么 google talk,msn,icq,facebook, 等等等等。所以，心有灵犀可以同时支持很多种聊天软件，不用再同时开着 gt，开着 msn，开着 qq 了，只要开一个 pidgin 就都能聊了。不过 qq 的信封比较特别，其他的聊天软件都使用公开的协议——至少能实现基本功能。有专门的文件写着自己只收什么什么样的信封，比如必须蓝色，上面印着蝴蝶，上开口直接撕开的信封才能发给 msn。可是 qq 这家伙的信封却很复杂，而且保密，别人都不知道具体应该怎么封。上面乱七八糟的有很多防伪标识，什么激光啊，磁条啊，比人民币还热闹。所为达到的目的就是只有自己的 QQ 软件能有互相通信。不过，强中自由强中手，人民币还有 HD90 呢，QQ 的信封怎么就不能伪造了？有牛人通过研究 QQ 的信封，慢慢分析，已经仿制出了 QQ 的数据包，可以实现最基本的文字聊天的功能了，所以您能看到心有灵犀也支持 QQ。但是这功能相当有限，只能是文字聊天，因为协议破解的不是很彻底。并且 QQ 官方指不定哪天就对协议做做修改，扼杀掉所有非官方的 QQ 客户端。所以建议您要是经常要用 QQ，还是不要用心有灵犀聊 QQ 了，聊 QQ 还是用官方提供的软件吧，那个软件，就是 QQ for Linux。

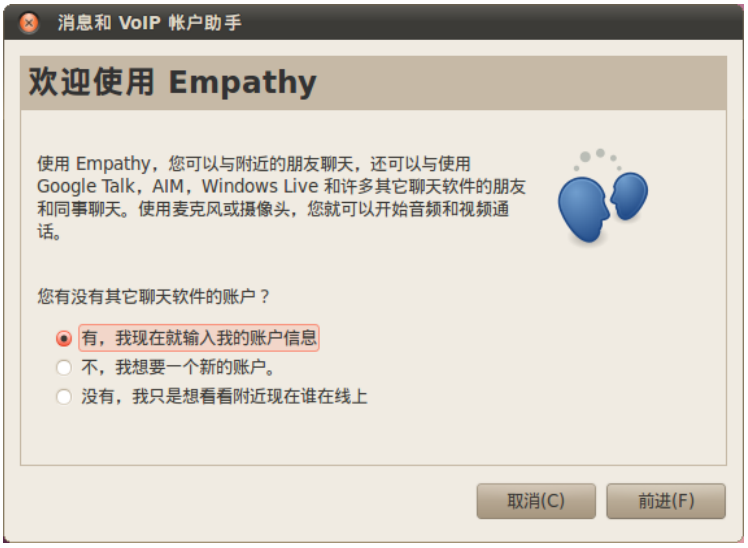
QQ for Linux 和心有灵犀一样，也是一个即时通信软件，也就是个送信的。不过您可别想着让超级牛力去装什么 QQ，他肯定告诉你：没有！因为 QQ 不在我们的软件源里，要想装，得让狐狸妹妹直接去 QQ 的网站上下载。这 QQ 是一个叫做疼痛，哦不对，叫疼……疼什么来着？哦，对，疼殉，一开始是疼，后来就殉了 -_-b，是一个叫疼殉的公司做的。话说这个疼殉啊，看人家 icq 软件玩的挺火，于是也弄了个 oicq，抢占了国内市场，结果一发不可收拾，毕竟那时候国内的网络还刚刚起步，这东西新鲜阿，于是全民 OICQ，大家上网聊天。后来 oicq 改名，叫 QQ 了，可是一直以来，由于各种原因，疼殉这个公司只能做出 windows 版本的 QQ 来，linux 下的没有。要说没有也不要紧，人家 google talk 也没有 linux 的版本，但是人家是基于开源的 XMPP 协议的，协议是公开的，于是世界各地的 linux 牛人们，很轻易的就做出了很多种用来在 linux 下聊 google talk 的软件。我们的心有灵犀，自然也是支持这种协议的，稍微设置一下就能聊 gatlk 了。可是 QQ 不一样，QQ 的协议是那个疼殉公司自己定的，还不让别人知道，又不提供 linux 的版本，结果，在 linux 下使用 QQ 一直是个很头疼事情。当然，这些都是历史了，现在疼殉公司终于想开了，提供了 linux 版的 QQ，虽然功能也很简陋，不过，文字聊天，传个图片，收个自定义表情什么的还是没问题的。要想装这个 QQ，就让狐狸妹妹去 http://im.qq.com/qq/linux/download.shtml 这个地址下载就好了。记住一定要下载最左边哪个 DEB 包的，那个是我们 Ubuntu 系统可以支持的，并且安装最为简单，双击就好。

QQ for Linux 虽然能用，不过也有不少的毛病。比如莫名其妙的经常占用硬盘，只要他开着，硬盘就经常工作着，也不知道读什么呢，所以很多人心里没底。于是就用网络版的 QQ 代替了。网络版的 QQ 也是官方的，所以不用担心哪天用不了，而且

使用也方便，不用装软件，只要用浏览器打开 web.qq.com 网页，登录进去就好了，功能比 QQ for Linux 还多。好，废话不多说了，主人已经点击了心有灵犀，我得赶紧叫心有灵犀起床去了。

心有灵犀来到工作间，这是他第一次运行，所以先很有礼貌的跟主人打了个招呼，介绍了一下，自己是用来聊天的软件，可以支持好多种协议，并且问主人：您有相应的帐号了么？

主人说：有！



好，那么心有灵犀又问：那您打算用哪个帐号聊呢？并且给出了一个下拉菜单，里面是他可以支持的所有协议。主人选择了 msn，并且在下面填写好了自己 msn 的用户名和密码。

心有灵犀再在窗口下面问：就光 msn 么？还需要不需要设置别的聊天帐户？主人想想：恩……需要。然后出现同样的一页，这回主人选了 google talk 并且填好用户名密码，然后说：就这俩了，没别的了。

心有灵犀继续问：您看，我还可以进行局域网通信，要是您内网里也有用同样软件的，就可以聊天了，就跟飞鸽似的，要不要开启这个功能呢？主人心说你怎么这么烦人呢～跟推销的似的。一想，我就这么一台电脑，根本用不着这功能阿，就一口回绝了。心有灵犀也是见好就收，再没废话了，赶快干正事。瞬间，主人看见了久违的 msn 和 google talk 上的好友，尤其是那个很聊的来的 mm，呵呵，赶快去打个招呼吧。

这一打招呼，坏了。

" 这是什么破输入法阿？怎么打字只能一个一个的蹦呢？ "

对于这点，我还是比较惭愧，说来还是我中文课没学好，写的不好不说（显示乱码），认起来也有问题（输入法不好用）。话说我们学校自从上一届开始，将 ibus 作为默认使用的输入法。这个 ibus 只是个输入法的框架，安装上什么输入法就可以用什么输入法。装上五笔就是五笔，装上拼音就是拼音，装上日文输入法就可以打日文了。ibus 是个比较新的输入法框架，所以有很多不成熟的地方，有许多热心的人为他开发输入法，光拼音就好多种。默认的这个不太好用，换一个就好了，换起来也简单：选择系统 --> 首选项 -->IBus 首选项，在弹出的对话框里面选择 " 输入法 " 标签，下面的 " 输入法 " 列表里面有很多的输入法，拼音输入法就不少。默认的那个看图标就不顺眼，可以选择那个 py，就是图标是一个拼字没有红 T 的那个。选中那个，然后点“向上”，把他放到最上面就好了。

主人就是这样的修改了输入法，可是换了之后还是一个字一个字蹦。为什么呢？主人很困惑，心说我给你一次重新做输入法的机会你不珍惜是不是？好，那我就……我就再给你一次。-_-b

于是主人又重新折腾了一遍，可是问题依旧，毕竟错误不是靠重复就能改正的，是不？最后主人终于爆发了，叫来新立得说：给我把 ibus 彻底删咯！得，看来 ibus 要被炒鱿鱼了。新立得立刻汇报（当然是问过超级牛力的）说：ibus 要删除的话，与之相关的 ibus-pinyin ,ibus-m17n 等等东西都需要删除。

主人毫不犹豫：那就都删！ 哎，看来是动真格的了。

新立得看看超级牛力，既然主人如此坚定，超级牛力也没啥说的了，找来 ibus，拍拍他的肩膀：兄弟，后会有期吧。

一转眼，就看见 ibus 拖着自己的行李——也就是他用的各种输入法插件，走了出来，一副垂头丧气的模样，看看我，再看看工作间里的同志们，摇了摇头，终于没有说出话来。后面跟着的超级牛力还不忘了嘱咐一句：东西都带齐了阿，本 APT 有超级牛力，也罩不主你了，路上小心点吧，这年头好多坏人拿着菜刀到处乱砍，尤其离幼儿园远点。一边说着话，超级牛力摇摇头，叹口气，用笔在他那小本上把 ibus 划去了。大家都为这么快就失去了一位战友而伤心，工作间里一片沉寂。这时主人发话：关机。哎，看来主人心情也不好，都没兴趣继续用电脑了。于是大家都默默的各自收拾东西，准备回硬盘睡觉。只有狐狸妹妹依然哼着小曲，保持以往兴奋的样子。哎，小姑娘不懂事，也不怪她。

重启之后，软件们各就各位，主人先叫来了新立得，下达命令：安装 ibus！呀？！！这是什么意思阿这是？新立得赶紧拍醒超级牛力，告诉他这个消息，超级牛力也摸不着头脑，但刚刚伤心的情绪却一扫而空，抱怨到：靠，这不是折腾我么，就说我有超级牛力也不能这么玩阿。一边说着，一边赶紧投入工作——查记录，汇报依赖关系，主人同意后上网，下载，转眼，ibus 又来到了我们面前。马上，主人又让超级牛力安装 ibus-pinyin——删的时候，ibus 一删，ibus 的所有输入法就都删了。可装的时候就得手动指定要装哪个输入法了。这回不知道是 ibus 痛定思痛，还是因为没有其他乱七八糟的输入法捣乱，总之，这回主人终于可以正常的打字了。我忽然想起了刚才狐狸妹妹的表现，问她：你是不是早知道主人要干什么？狐狸说：嘻嘻，那当然了，主人就是让我去网上找的解决方法，看到有人说把 ibus 重新装一边就好了，才这么干的。

经过这一场风波，ibus 更加的珍惜这个工作机会，毕竟自己混这一行的日子还不长，和其他几位老大比起来，像什么 scim 阿 fcitx 阿，自己还是个输入法界的小弟，还有很多东西要学习。

scim 是以前我们 ubuntu 默认使用的输入法，我的学长们，像 9.04，8.10 他们，以及更早的版本，都是用 scim 的。从 9.10 开始才换成了 ibus。所以，有不少人还不习惯 ibus，想要换回 scim。要想使用 scim 做输入法，首先第一件事当然就是安装 scim 了，这个不必多说，找超级牛力和新立得就对了。scim 也只是一个输入法的框架，具体要用什么输入法，还得装各种插件。scim 有些年头了，有许多支持他的优秀的输入法引擎，比如用的最多的智能拼音，SUN 拼音，Google 拼音等等，用什么引擎装什么。比如要用拼音，就得装 scim-pinyin。装好之后还不算完，因为这时候系统默认的还是 ibus，得把 scim 设置成系统默认的输入法才行。这个也简单，点：系统 --> 系统管理 --> 语言支持，下面，输入法那选择 SCIM 就好了，要换 fcitx 也是一样。我的主人没有换输入法，他似乎觉得 ibus 还是挺称职的。

2.4 多媒体

主人终于和 mm 幸福的聊在了一起。主人首先很高兴的显摆了一下自己是在一个叫做 Ubuntu 的系统下。mm 一脸迷茫（虽然看不见，但我估计是），于是主人详细的解释了 ubuntu 到底是什么，Linux 是什么。但是 mm 似乎对这些并不感兴趣，于是岔开话题，两人聊起了音乐。最近 mm 在听一首歌，感觉很不错，就介绍给主人。主人自然愿意主动贴近 mm 的生活，于是，叫来狐狸妹妹，上网去找这首歌曲。

很快，狐狸梅没找到了这首歌，并且下载了下来，是一个 mp3 文件。主人很自然的就双击了这个文件，想要播放来听听。

播放音乐这种事，要找我这里的多媒体部门，“电影播放机”就是其中的一员，他的英文名字叫 totem。于是，我看到主人要播放 mp3，就赶快叫醒了 totem 来干活。totem 一本正经的拿着这个文件翻来覆去的看了半天，摇摇头说：这个……播不了。我一听就急了：你不是播放机嘛？在学校里你天天的吹牛说你什么都能播，视频音频通吃，怎么这会掉链子了。totem 赶快解释：老大，别急别急，我说播不了，是有原因的。我播音频也好，视频也罢，都需要解码器。我现在手头没有这个 mp3 的解码器，所以不能播放。

有人问：解码器是干什么的？要知道，音乐也好，视频也好，他们的格式有很多种，就好象现实中看电影，有数字电影，就要用数字放映机。胶带的，就得拿传统的放映机。在家里看光盘的，就得拿 DVD 机，看录象带的，就得拿录像机。听音乐也是，磁带的和 CD 的，肯定没法都塞进同一个机器里。

Totem 这样的媒体播放软件就像个电影放映员，解码器就是放映

机。放映员在怎么牛，也得有各种放映机做支持，没放映机他啥也放不了。我刚想质问 Totem 为什么不把解码器带全了，转念一想，我自己也没把语言支持包带全嘛。算了，还是不追究这个了，先说眼下怎么办吧。totem 说：没关系，没带就让超级牛力去装嘛。当然，装软件这么重要的事情，我们得跟主人请示一下。于是，totem 在屏幕上显示出了一条信息，大概意思是，您要播放的这个东西所需要的解码器似乎没有安装，请问要不要先在去搜索一下需要装的解码器呢？主人怎么样，主人还有选择么？不听 totem 的这 mp3 就播放不了阿，所以点击了“查找”。

然后 totem 开始搜索，很快得出结论：您需要安装 gstreamer0.10-ffmpeg，gstreamer0.10-fluendo-mp3，gstreamer0.10-plugins-ugly 这三个包，我会让超级牛力去办理，您看怎么样？

主人当然点击了“安装”。之后，估计您也猜到了，当然是超级牛力开始工作，装上了需要的解码器，最终 totem 终于发出了久违的绵羊音——绵，绵羊？！

主人显然和我一样对这个音乐不感兴趣，不过却对 totem 很感兴趣。因为以他以往的经验，系统刚刚装好，不能播放某音频或视频文件是正常的，比如查皮刚装好时，没有软件可以播放 rmvb，那就必须得装软件，而装软件这个工作需要用户自己去做，上网找，下载，搞不好下载回来是个压缩包，那还得先装解压缩软件。可是 totem 竟然不用如此繁琐，直接提示缺什么东西，只要点个确定，该装的就都装上了，太人性了。于是主人很兴奋的又找来个 rmvb 双击了一下，totem 立刻换过视频解码器，开始播放 rmvb 的视频。

虽然 totem 能够播放不少类型的视频，不过毕竟他只是多媒体部门的小弟，要说播放器，那还得说是 Mplayer 老爷子。Mplayer 可是多媒体部门的元老了，能力相当强，什么片都能放，什么 rmvb，flv,avi，wmv 全都不在话下。（当然，前提还是得有解码器）就算您没图形界面，人家跟字符界面照样能依靠 framebuffer 给你放电影，甚至还

能给您拿字符拼出电影看。现在时代发展了，都高清了，人家也不甘落后，照样能支持，什么硬解码软解码的，通吃。主人也在网上听说了 mplayer 老先生的名声，于是叫来新立得，说我要装一个叫作 mplayer 的软件。新立得自然是告诉超级牛力，超就牛力自然又一次完美的完成了工作。（貌似我们这就超级牛力最忙了）装完了之后，主人杯具了——装哪了阿？

这里，要介绍一下我们 Linux 和那个查皮的不同哲学了。我们两个的做事方式，对待用户的态度，以及很多观念都有很大的不同，这主要是由于查皮是个闭源的系统，而我是开源的系统。

什么是闭源呢？就是源代码不开放。我们知道，程序是程序员们一行一行的语句编出来的，c 语言也好，java 也好，这一行一行的语句，就是这个程序的源代码。有了源代码，就能够 100% 的了解整个程序的构造，如何工作。而源代码是不能运行的，必须要把源代码变成可执行的二进制程序，这个过程叫做编译。源代码经过编译之后，才可以运行，但是编译之后的程序就不能够知道内部的构造了。我们平时在网上下载的各种程序，都是编译好的二进制程序，如果你想要它的源代码，对不起，不行！这是商业秘密，怎么能给你？给了你，我们的软件怎么卖钱？这种不开放源代码的程序，就叫闭源程序。打个比方，就好像肯德基。香辣鸡翅谁都可以得到，只要花钱买就行，但是配方没人知道（虽然其实也没多好吃吧）。配方就相当于源代码，香辣鸡翅就相当于编译好的二进制程序，制作过程就相当于编译过程。如果有了配方（源代码）你就可以自己作香辣鸡翅（自己用源代码编译出二进制程序），甚至还可以根据口味对配方进行修改。（根据自己的需求修改源程序，为软件增加自己需要的功能）

那开源又是什么？开源是一种精神，是乐于分享的理念。再举个例子，有一天你发现，蒸鸡蛋羹的时候往里面加点牛奶，可以让鸡蛋羹更滑嫩。知道了这个窍门，你很高兴的把它告诉你的朋友，让他们分享你的经验，于是大家很高兴的也学会了做这样的鸡蛋羹。这就是开源。你也可能不把它告诉别人，而是保留这个秘密，甚至申请个专利，然后开个店去卖京城独一份的奶香滑嫩鸡蛋羹。这就是闭源。当然，这之中没有谁对谁错，谁好谁坏，只是不同的理念而已。

正由于查皮闭源的天性，导致他不希望让别人了解自己的结构，所以查皮下的系统文件都是统一放在一个目录里面的。一般叫个什么 windows 阿之类的目录。里面的东西都是查皮自己的，别的软件谁也别动。有什么样的领导，自然就有什么样的员工。查皮下的其他软件也都学查皮那样，给自己建一个目录，跟自己有关的东西就都放在那个目录下。软件之间泾渭分明，互不干涉，老死不相往来。顶多了早晨上班见面点个头而已，很少有其他的交流，工作间里一片死气沉沉。

而我们 Linux 的世界就不同了。

我们这里的目录是开放式的，按照文件的用途划分各种目录，而不是按照软件的名字来分。每个软件都把需要用的文件放在公共的地方，如果别的软件也需要用，甭客气，拿走。就比如刚才 totem 装的那些解码器，其实就是一些解码用的库文件，所以放在了 /usr/lib 下。现在来了 mplayer，他要想播放视频也需要解码器，如果是在查皮下，那就请您自备一套，甭想用我的。就像 windows media player 不能播放 rmvb 格式，装了 realone 后，realone 可以播放 rmvb，但是 windows media player 照样不能，因为 realone 带来的解码器只能 realone 用。我们这里就不是这样，既然有解码器了，就大家一起用。mplayer 抄起 totem 刚才装的解码器就能播放视频，这样，避免了同样功能的解码器重复开发，也省的为某一种视频格式专门装个软件。

说了这么半天，您一定着急了：“这 Mplayer 到底装哪了阿！！”对于这个问题，我只好告诉你：“就像一锅被火车撞了的豆腐脑一样散落在各处了……”二进制文件放在了 /usr/bin, 库文件放在了 /usr/lib，配置文件放在了 /etc/，其他一些文档文件放在了 /usr/share，还有一些数据放在了 /var。乱吧，你肯定会说，这么乱，怎么管理阿！要删除的时候怎么办？这个问题的答案是：相信超级牛力！他会记得这些安装过的文件，删除的时候肯定一个不留。

那既然安装了 mplayer，到底怎么运行他呢？如果是安装了一个图形界面的程序，我会按照这个程序的类型，自动把这个程序放在“应用程序”菜单里。不过 mplayer 是个字符终端的程序，所以就没有在图形界面体现了。要想运行，打开终端输入 mplayer < 要播放的文件 > 就运行了。这么运行当然有些麻烦，所以，主人很快又叫来新立得，安装了 gnome mplayer。这又是什么呢？这是个 mplayer 的图形化前端，可以像 totem 一样的操作，而后台还是 mplayer 在做实际的播放工作。除了 gnome mplayer，还有 smplayer，kplayer 等等很多 mplayer 的图形前端。这些个 player 当中，要数 smplayer 功能最多，使用最方便了。

mplayer 和 totem 虽说音频视频都是可以播放的，但毕竟视频才是主业，要说听歌的话，还是得找专业人士，比如我这里的 Rhythmbox 就是个不错的音频播放器。Rhythm 是节奏，韵律的意思，所以我们就叫他韵律盒吧。

他可以播放各种音频文件（他也是依靠 totem 装的那些解码器来播放的），还可以收听网络电台，还可以去 ubunt one 网络商店购买歌曲来收听，总之功能很强大。主人经过摸索，终于在应用程序 --> 影音菜单中找到了 Rhythmbox 音乐播放器，赶紧把他叫起来干活，找来刚才那个 mm 发过来的文件（没办法，系统里暂时没有别的音频文件），让韵律盒播放试试。这对韵律盒自然不是难题，马上，工作间里就到处咩咩声不绝于耳。主人还没来得及高兴，就发现问题了，怎么这歌曲名又乱码了呢？

刚装上没中文，后来有中文了但是 Flash 乱码，先在 Flash 没问题了，听个 mp3 还

是乱码，真是乱码年年有，今天特别多。这 mp3 的乱码又是怎么回事呢？

话说一个 mp3 的歌名阿，作者阿，是记录在 mp3 标签里面的。这个标签的记录方式有很多中，主要是这么几个标准：ID3v1, ID3v2 2.3，ID3v2 2.4，APEv2。这个 ID3v1 的标签支持 ISO-8859-1 编码，这是个国际通用的编码，但是很可惜，他是不支持中文的。到 ID3v22.3 这个版本开始，增加了对 utl-16 的支持，utf-16 也是一套国际通用的编码，这里面就包含中，日，韩等等各国的文字了。到 ID3v2.4，更增加了 utf-8 的支持。utf-8 和 utf-16 都是国际通用的一套编码，所不同的是 utf-8 以字节为编码单元，而 utf-16 用双字节为编码单元，这样就有大小端问题（也就是哪个字节先传输的问题，不知道的可以 google 一下）。因此总的来说，utf-8 相对更受欢迎一些。ID3v22.4 支持 utf8 编码，可并没有说一定要用，只是可以用 utf-8，自然也可以用其他的编码。而 APEv2 标准就不同了，它规定了其编码必须同一为 utf8 编码。

ISO-8859-1，utf8，utf16，这些都是国际标准的编码。但 utf8，utf16 的出现都是后来的事情了，一开始，是没有国际标准的中文编码的，那时候只有国标——国家标准。也就是我们常见的 GB2312，GBK 和 GB18030。由于这些只是国家标准，所以开源软件的作者们（多数不是中国人）自然是忽略的这些标准（当然了，每个国家都有自己的标准，听谁的阿？）。所以，如果是用 ID3v1 或者 ID3v2 类型，并且使用国家标准编码的 mp3 文件，就会出现乱码。如果是 ID3v1 或 ID3v2 类型，但是使用 utf-8, 或者 utf-16 编码的，就不会乱码。如果是使用 APEv2 标准的 mp3 文件，就更不会乱码了，因为 APEv2 必须用 utf8 嘛。不过遗憾的是很多播放软件不支持 APEv2 标准，好在我们的韵律盒是支持的。

然而事实是残酷的，事实就是：国内的大多数 mp3 都使用了国家标准编码。那我们在 Linux 下就看不到了么？当然不是，编码不对的话，转换一下就好了嘛。有个软件，叫做 mid3iconv，他就认识各种编码，让他把 mp3 的编码改成 utf-8 的，就可以了。这个软件从哪里来？自然是让超级牛力请来：sudo apt-get install python-mutagen，或者让新立得安装 python-mutagen，一样。装完了以后，就可以在你的家目录下运行：

```
find . -iname "*.mp3" -execdir mid3iconv -e gbk {} \;
```

所有的 mp3 就都改过来了。

有人说，为什么查皮那里的播放器就没这么多问题。人家毕竟是商业化的产品嘛，而且是专门的“中文版”，自然得入乡随俗的支持国家标准编码。查皮下的播放器默认的有个叫 WMP 的，WMP 支持的 mp3 标签类型其实也不多，他不能支持 ID3v2 2.4 和 APEv2 的标签，还不如我们的韵律盒支持的多。但是他很聪明，不支持的标签我就不显示，以显示那个 mp3 文件的文件名来代替，这一点倒是值得我们的软件们学习。

2.5 硬件

到目前为止，主人的 Ubuntu 体验还算愉快，这主要是由于这个电脑上的硬件我都可以比较顺利的驱动起来。比如网卡，能够很顺利的连上网络，能够让超级牛力之类的软件发挥出最大的作用是很重要的。如果网卡不能够驱动，那可就麻烦了，要知道，一个离开了网络的 Ubuntu 基本就是个废物。所以，硬件驱动很重要，我们 ubuntu 也是意识到这点，正在努力的支持更多的硬件，默认支持更多的驱动。有人问，什么是驱动？

这个电脑硬件阿，不像电视机电冰箱似的，买来插上就能用。硬件要想在计算机上工作，得需要会操作它的软件，这个事情，一般就是归我们操作系统管了。But 子曰：“人非生而知之者，孰能无惑？”这时候狐狸妹妹轻蔑的哼了一下说：“那不是子曰的，是韩愈曰的”好吧，不管是谁说的，反正道理是这个道理，我们操作系统，也不可能生来就会操作所有的硬件，就像你不是生来就会开飞机一样，得学，得考本，得移库、倒库、坡起、限制门。狐狸妹妹再次严厉的鄙视了一下：“你见过飞机过限制门么？！那是汽车。”反正，我们要想会操作一个硬件，也需要学习，这就需要驱动程序，任何硬件要想工作都是需要驱动程序的。这时候可能有人提出反对意见了：“硬盘，光驱，这些也都是硬件哪听说过要装驱动程序的？还有我的 U 盘，摄像头，也都是插上就能用，不用装驱动阿。”不用装驱动，不代表不需要驱动。硬盘光驱是最基本的存储设备，在我们操作系统起床工作之前，

硬盘就要工作（因为我们都住在硬盘里嘛，必须硬盘工作，我们才能被读进内存里。）这个时候其实是另一个软件——BIOS 在操作硬盘工作，硬盘的驱动，就在 BIOS 里。关于 BIOS 这老人家，我们以后细说。反正硬盘光驱这样的基本设备的驱动很简单，也统一，任何一家生产的硬盘都是一样的用法，所以硬盘光驱的驱动就被集成在了 BIOS 和操作系统里面，不用额外安装。其他所谓不用装驱动的设备也一样，都是因为驱动集成在了系统里。比如查皮他们家以前的瘟酒吧系统，就不认识 U 盘，需要装驱动才行。到查皮这一代，就不用装了，集成了。

驱动就像一本给操作系统看的使用手册，上面写明了如何如何操作这个硬件，写哪个寄存器就把数据发出去了，从哪个寄存器读就把数据读回来了，往哪个寄存器写个什么什么数据就自爆了等等。（这是什么硬件阿……）就像买来电视机，里面的使用手册一样。针对不同的操作系统，需要有不同版本的驱动程序。这个好理解吧，因为我们是完全不同的系统嘛。

查皮和他以前的酒吧呀，之后的喂死它，他们都是有点软公司的系统，他们能用的驱动还不同呢，我们这根本就不是一个阵营的，那就更不一样了。我们和查皮就像说着不同语言的不同国家的人。我们能看得懂的手册，查皮看不懂，反过来也一样。你家电视机的说明书不也有中文版，英文版，韩文版，非洲土著语版么。但是，并不是每个硬件厂家都给每个系统制作一份驱动的，毕竟厂商人力财力有限。电视机也不是每个都有非洲土著语版的说明书嘛。（压根就没有把……）所以，一般硬件厂商会优先开发市场占有率最高的那个系统的驱动程序，哪个系统？目前来说，就是查皮和他的后代喂死他、温妻了。而我们 Linux 就经常遇到一些无法使用的硬件，很多人还抱怨我们无能，冤枉阿～～～

其实我们 Linux 能够支持的硬件已经逐渐多起来，大多数主流的设备基本不用装驱动就可以使用了。一般像我们 Ubuntu 吧，装完了系统之后也就装装显卡驱动就可以了，没准连显卡驱动都不用装。比如我住的这台电脑，用的是 Nvidia 的显卡，这家公司对我们 Linux 还算比较友好，提供了不错的 Linux 驱动。

这不，现在主人就想装显卡驱动了。装显卡驱动说容易也容易，说麻烦，也挺麻烦的。说容易呢，你点击系统 --> 系统管理 --> 硬件驱动。如果里面有你显卡的驱动，选中，启动，就好了。当然，免不了重启一下。说麻烦，如果这样作不行的话，那就麻烦点了。得看你用的是什么显卡，要是 Nvidia 的，那就好办点，要是 ATI 的，那就更加麻烦，不过也还算有方法。要是什么 sis，via，那能够正常显示出桌面就不错了，凑合用着吧。主人这台电脑用的是 nvidia 的显卡，属于最好办的那种，官方提供的驱动就不错。

主人发话，狐狸妹妹立刻就去官方网站上找到了那个驱动的安装程序，然后一个媚眼就把那小子领回了家里，小子名字还挺长，叫什么 NVIDIA-Linux-x86_64-190.53-

pkg2.run，从名字看是专门给我这样的 64 位 Linux 服务的。回来之后主人先是让超级牛力删了原来的驱动（就是通过”硬件驱动”安装的那个，如果没装自然就不用删了）——怕有冲突，两本手册都叫“xxx 显卡驱动”，万一我一忙乎拿错了不就乱了么。删了之后主人双击那个新来的驱动安装程序，让他运行，结果那小子派头挺大，跟主人说“你这个……图形界面还开着呢，没法装，先把图形界面关了再找我！”主人无奈，只好 ctrl-alt-f1 进入了字符界面，登录之后运行 sudo /etc/init.d/gdm stop，意思就是告诉图形界面那哥几个，回硬盘歇着去吧，暂时没你们事了。顺便插一句，其实我还是喜欢主人用文字跟我交流，有种平等的，倾诉心声的感觉，比在图形界面里被指来指去的舒服多了。图形界面哥几个彻底休息之后，主人又运行 /< 路径 >/NVIDIA-Linux-x86_64-190.53-pkg2.run 叫醒那个安装程序，这回小子又把嘴一撇：“你是 root 么？不是 root 没资格跟我说话！”气的主人抓耳挠腮，只好乖乖的在命令前再加上 sudo，这回那小子终于运行了，先是叽哩咕噜的说了一大堆英语，好象是问主人要不要去网上下个啥东西，主人毅然决然的回绝了他。之后的过程很顺利，小子仔仔细细的吧这里检查了一遍，迅速编写出一本我看得懂的模块塞到我的手里，又改好了 xorg 老大需要用到的配置文件，然后向主人汇报：“行了，没事了，重启图形就好了，我睡去了。”

之后，主人又用 sudo /etc/init.d/gdm start 叫醒了图形界面那帮倒霉的孩子们——被子还没铺好呢又被叫出来了。然后，漂亮的图形界面就回来了，原来不正常的分辨率，这回可以调节正常了，原来不能启动的桌面效果，这回可以启动了。

那么如果是 ATI 那家的显卡呢？据说就不这么容易了，不过其实我也没见过，只是这么听别人说，听说他家的驱动要靠人品，呵呵。再有比较好的就是淫特二那家的集成显卡，他家的显卡驱动直接贡献给我们学校，我们走出学校的时候就自备他家的显卡驱动模块了，而且支持的挺好，不用再安装了。

Chapter 3

第三章

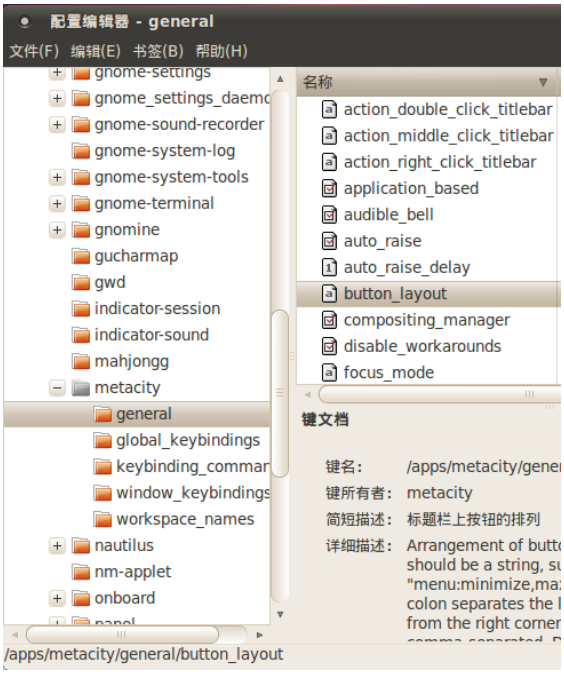
我的系统 我作主

3.1 我的桌面

经过了短暂的磨合，通过主人对我们的各种配置，总算是可以满足主人的初步需求了。不过这自然还不算完，默认的样子并不是每个人都用着顺手的。

比如说默认的图形界面主人看着就不顺眼，为什么呢？因为窗口上的关闭，最大、最小化的按钮竟然在左边。这让主人很不爽，每次要关窗口的时候鼠标都会划出一个纠结的曲线——先向右上方去，发现按钮不在原来的位置之后又打轮拐向左上角来。听说今年我们这界的图形界面组的那哥几个都是种苹果的出身，所以审美观点可能跟以往有些不同，不过没关系，我们 Linux 是为人类服务的，哪里看着不爽都可以改嘛。

主人上网学习了一下之后就开始动手了，首先启动命令行，并且运行 gconf-editor，或者按下 Alt-F2输入 gconf-editor 也可以。gconf-editor 这家伙是一个 Gnome 桌面环境的配置记录员，有点像查皮那里大名鼎鼎的注册表。不过查皮的注册表是用来管理整个系统的，而 gconf-editor 只管 Gnome 桌面环境的事，别人他管不着。gconf-editor 启动之后，主人找到了 apps/metacity/general，打开里面的 button_layout 条目。把里面的内容改成了 menu:minimize,maximize,close。然后一点确定，窗口上的按钮马上就回归以前的风格了。



窗口按钮顺手了之后，主人又开始追求更加新奇的东西——3D 桌面。

这个3D 桌面的概念，说起来是那个有点软公司先想出来的。那时候有点软公司正在研制那个"喂死它"系统（听着像灭蟑螂的……-_-b），作为重要特性之一，"喂死它"系统支持三维桌面，当时还流传着一张图，展示了切换桌面窗口的时候以立体的形式展现每一个窗口。这样新颖的设计着实让人们眼前一亮，也让 Linux 界的高手牛人们心头一热——这么个效果，我们也能做出来阿。

要想实现3D 的桌面，那么原始的 xorg，就不能够胜任了。一小搓明白此真相的牛人们本着重新发明轮子的精神，设计了新的 X 协议实现，用以代替 xorg 的，这就是 XGL。有了 XGL 作为 X Sserver，加上 Opengl 的支持，上层的软件就可以比较轻松的绘制出立体的窗口，于是在此基础之上，Compiz 诞生了。刚刚出世的 Compiz 功能不多，只能够让窗口随意透明，有拖动窗口时的果冻效果，以及最大化最小化时候的缩放效果，还有切换桌面时候的立方体效果。但就是这些，已经足够让 Linux 的用户们大为惊艳，让 Windows 的用户们极度震撼。那时候"喂死它"还在开发，还仅仅能够通过几张图片

和几段视频来获知他所谓的3D 桌面，而这时候 Linux 下已经可以看着一个硕大的立方体转来转去了。

XGL+Compiz 的成功吸引了更多的人加入到完善 Linux 的3D 桌面的工作中去。其中，xorg 的用户和开发者们都在想一个问题——xorg 为什么不能实现3D 桌面呢？为什么非要换成 XGL？通过思考，最终他们得到了答案——不用非得换成 XGL，xorg 就可以！开源的优势这时候体现了出来，xorg 是开源的，大家都可以来对他修改，于是有人为他设计了 AIGLX 插件，有了这个插件，xorg 可以和 XGL 一样实现3D 桌面的绘制工作，甚至比 XGL 还要可靠。说来也是应该的，毕竟 xorg 作为标准的 X Server 工作了那么多年了，多年的沉淀得到的稳定性不是一朝一夕可以追赶上的，带有 AIGLX 插件的 xorg 逐渐替代了 XGL，到现在，xorg 依旧是标准的默认的 X Server，要实现3D 桌面不再需要安装 XGL 了。

在底层的 X Server 发展的时候，Compiz 也没有闲着，功能在逐渐的完善，一方面更加的稳定，效果更加流畅，另一方面更多新的效果被开发出来。可是渐渐的，追求稳定的开发者和追求效果的开发者分歧越来越大，最终导致 Compiz 项目组的两部分人分道扬镳。一部分人继续维护着 Compiz，不再增加什么绚丽的特效，而是注重稳定性和对资源的利用。另一部分人离开了 Compiz 项目，出去自立门户，成立了 Beryl 项目。这个项目旨在设计出炫目新颖的桌面特效，让 windows 界的看看，他们想做的，我们一样能做出来，并且还比他们做的好。从此，什么火焰，魔法，神灯，爆炸，屏幕涂鸦，飘飞雪花，各种特效层出不穷，真是千变万化，应接不暇。

然而有道是：天下大事，分久必合，合久必分。经过了一段时间的发展，Compiz 项目逐渐稳定，基本生没有什么 bug 了，没什么目标的程序员们也开始想在 Compiz 里加入新的效果。而 Berly 在经过了创造的高峰期后，也没有什么更新奇的特效设计出来了。于是，两方人马不约而同的向着昔日的伙伴深情一望——咱还合并吧。结果，神机百变的 Beryl 和老成练达的 Compiz 终于又走到了一起，两个项目合并为了 Compiz Fusion——也就是现在我们 Ubuntu 系统中用来实现3D 桌面效果的软件。

其实在3D 桌面非常成熟的今天，我们 Ubuntu 已经默认自带一些简单的效果，大约就是当年 Compiz 最初时期的那点。想要体验的话，首先要安装好能够支持 Opengl 的

显卡驱动，之后只要在系统-->首选项-->外观 里面的“视觉效果”标签里，选择“正常”，或者“扩展”就可以看到了。扩展里面的效果自然要比正常多一点，不过也是很有限，如果想有更多的效果，并且可以自己配置，那么需要像我主人这样，安装 Compiz 设置管理器，安装也简单，叫超级牛力：sudo apt-get install compizconfig-settings-manager 就好了，或者让新力得装当然也行。装好之后就可以在系统-->首选项 里，看到“compizconfig 设置管理器”。运行之后，就可以看到有很多效果可以选择，之后，就根据自己的喜好搭建你自己的桌面效果吧。

故事外的事——Xorg

这一节在咱们说3D 桌面提到了 Xorg，这 Xorg 是个什么软件呢？

话说我们 linux 系统里，系统跟图形界面是分开的。绘制图形界面的事情由专门的图形部门来负责。而图形部门的老大，就是 Xorg。他会跟硬件打交道，会用显卡（当然，用显卡也得经过我），能在显示器上画东西，想画什么画什么，谁要想显示点东西给主人看，都得经过他。 要想跟 Xorg 打交道，在显示器上显示出图形来，得懂他们图形部门的黑话——学名叫协议。他们说话使用一种叫做 X 的协议，这个 X 不是牛 X 的 X，也不是傻 X 的 X，而就是 XYZ 的那个 X，XP 的那个 X，反正就是个 X。一个程序通过 X 协议告诉 Xorg 要画什么。那么这个程序就是 X Client，而 Xorg 就是 X Server。反正要想显示图形，就得用这种黑话跟 Xorg 去说。每一个要显示图形的程序都得会这种黑话，比如狐狸妹妹，要显示东西，就说：“驼子碗，筛土的抛闪！”那意思就是说画一陀黄色的便便-_-b，当然，这就是比方，其实我可不懂他们的黑话。（这一点不像查

皮，他本身兼职负责画图形）那么程序要画什么直接跟 Xorg 说就行了么？其实也行，比如 Mplayer，他可以直接跟底层的 X Server 交流，可那就像是在字符界面下看片了——没有窗口，图像没法移动，没法全屏，没法最小化等等。因为 MPlayer 只负责放片，像画窗口啊，移动窗口什么的这些事情他可不管。那谁来管呢？这时就需要一个窗口管理器，我们这里默认的窗口管理器叫做 metacity（就是 Gnome 下的默认窗口管理器）。程序要画什么东西其实是跟他说的，比如 Mplayer 说：“画一只猪 ”（当然是用 X 黑话）于是 Metacity 转头告诉 Xorg：“在某某位置画个方的窗口，在里面画一只猪。”过一会可能主人觉得 Mplayer 的窗口挡着他和 MM 聊天了（那是，猪哪有 MM 好看呀），就把 Mplayer 的窗口挪了挪，于是 Metacity 又对 Xorg 说：“把刚才那只猪和窗口往左移动3.2厘米。”这个过程 Mplayer 是不知道的，他只管专心的向 Metacity 描绘着影片中的一幅幅图像：“猪，走 路的猪，跑动的猪，跌倒的猪，捆绑的猪，烤熟的猪……”

3.2 我的网络世界

除了桌面以外，软件也得找顺手的。就说这上网吧，主人就总觉得这狐狸妹妹用的不大顺手，因为他以前在查皮那里是用一个叫做 Chrome 的浏览器来上网的，那个浏览器简洁方便速度快，价格便宜量又足，主人一直都用他，这忽然一换成狐狸妹妹，还是有点不大习惯。既然不习惯就换嘛，所以主人想在我这里安装个 Chrome 来用。好在这个 Chrome 是狗狗那个公司做的，这个公司还是比较照顾我们 Linux 的，很多软件都有 Linux 版，这个 Chrome 也不例外。于是主人就牵着狐狸去狗狗那里找到了 Chrome 的 Linux 版。点了下载后，那网站给出了几种版本：有 rpm 的，有 deb 的。这 rpm 的版本是用在另一类基于 Redhat 的 Linux 发行版上面的，像 Suse, Mandriva, Centos 之类的。我们 Ubuntu 是基于 Debian 的，所以必须装 deb 包的版本。deb 版本又分为32位和64位，这说的是操作系统的位数。由于我是64位的系统，主人自然选择了64位的 deb 包来下载。（当然，可不是说 Ubuntu 都是64位，Ubuntu 有64位版和32位版，看你装的是哪个了。）狐狸妹妹开启了一个下载进程，不急不忙的下着，看得主人有些不爽：当年在查皮的时候下载起来有雷有车，如今到此地界，难道就只能靠这浏览器自带的下载功能？ 赶紧去网上问问，才得知，这火狐狸本身虽然之是个浏览器，但是她可一安装很多强大的扩展程序，装了不同的扩展，就有了各种不同的功能。主人对此很感兴趣，于是赶紧先去找个下载用的扩展来装装。

给狐狸妹妹装扩展很简单，跟用新立得装软件差不多，也是“超市化”安装。打开狐狸妹妹后，选“工具->附加组件”，就会看到安装附加组件的窗口。在上面的搜索栏里面输入你要查找的插件名字，找到后就点"添加至 Firefox"就好了。

当然你可能不知道到底要装什么插件，也不知道到底都有什么插件，没关系，点那个“浏览全部附加组件”，点击之后会打开一个网页，就是这个地址：https://addons.mozilla.org/zh-CN/firefox/。在这上面分门别类的介绍了现在流行的，

好用的，各种各样的组件。看到合适的插件，想要安装的话，直接点击网页上的按钮就可以装上了。主人之前已经通过在论坛询问得知狐狸有一个扩展组展叫做 DownThemall，是用来下载的，于是就直接搜索这个组件，告诉狐狸妹妹“装！”狐狸妹妹赶紧去下载这个扩展，下载来之后把这个扩展保存在硬盘里属于她的那块空间中，然后告诉主人：重启我一下就可以用啦。 主人点点头：“好，重启吧。”于是狐狸妹妹回硬盘，收拾了一下，再次启动来到内存——这回她兜里就插上了 Downthemall 扩展了。这回主人再点击任何下载的链接，就可以选择用 downthemall 下载了。这样的好处主要是可以多线程下载，提高下载速度。但是这只限于普通的 http 或者 ftp 下载，什么迅雷专用链接，快车专用地址之类的自然就没戏了。bt 阿电驴阿啥的也是有其他的专门软件来负责，Downthemall 只负责最基本的下载，不过目前来说，这就是主人想要的，足够了。

装上了 downthemall 扩展，主人用着觉得还不错，于是又在网站上面浏览其他的扩展软件。这里的扩展软件还真是琳琅满目，干什么的都有。有用来预报天气的，有方便管理 blog 的，有网页开发相关的，有阅读新闻的，有下载的……等等等等，真是没有做不到，只有想不到。

主人先是看到了一个叫做 1-ClickWeather 的扩展，这是个天气预报的插件，主人一想……这浏览器带天气预报，挺有意思，就顺手在网页上点了旁边的"添加到 Firefox"，然后自然是狐狸妹妹下载这个扩展，把这个扩展保存在硬盘里属于她的那块空间中，然后再告诉主人：重启我一下就可以用啦。 主人又点点头：“好，重启吧。”于是狐狸妹妹还是回硬盘，收拾了一下，又一次启动来到内存——这回她兜里就插着 Downthemall 和 1-ClickWeather 这两个扩展了。好在扩展一般都不大，几百 K 而已，下载起来快，揣在狐狸身上也不沉。这回狐狸刚一启动，就先弹出了一个用于设置1-ClickWeather 的窗口。

因为是天气预报嘛，世界这么大，不能都给你报，你得选择一下你要看哪的天气预报。主人输入了 Beijing，狐狸妹妹拿着扩展查了一下，问是不是 Beijing, China 这个地方？主人说没错，就是这个。然后，狐狸就正常的启

动了，启动后就可以在右下角看到最新的天气预报。

除此之外，主人还看上了 DownloadHelper 扩展。这个扩展对主人也很有用，平时他经常上一些地瓜啦，马铃薯啦之类的网站，看些有意思的视频。不过这些网站都是在线看的，不提供下载。主人的网速不是很快，再说，就算网速足够快，也不能保证什么时候想看就能上去看，今天还有的视频，明天就可能被大螃蟹抢走。（你懂的，是吧）所以主人希望能够把网页上的视频保存下载，一看到这 DownloadHelper 扩展，暗自叫一声正合朕意，赶快让狐狸妹妹去装。狐狸妹妹二话不说，责无旁贷，赶紧下载，保存，重启，揣着三个扩展进内存，一气呵成。这回如果网页上再有什么 flash 的视频拉，音频啦，狐狸妹妹就会通过 DownloadHelper 这个扩展检测到，并且通知主人，有个视频可以下载。如果主人需要，就可以把这个视频下载到本地硬盘上了。

之后，主人又装了7，8个有意思的扩展。装了这一大堆扩展后，虽然把狐狸妹妹累的不轻，启动都变慢了。但主人却开始对狐狸妹妹有好感了。体会到了狐狸妹妹最大的特点——定制，就是可以变成你想让她变成的任何样子，按照你的口味，打扮起来给你看。不过有时候也被人诟病说这么多扩展拖慢了速度，这方面确实不如 Chrome。哦，对了，主人那 Chrome 早就下完了，狐狸妹妹也提示过他了，他到底装不装阿。

主人和狐狸妹妹又缠绵了一阵子，终于想起来了——哦，对了，我是想装 Chrome 的,明明是要换用自己顺手的浏览器，怎么差点被狐狸迷惑了。他赶紧找到下载的 Chrome 的 deb 包。由于主人没有更改保存地址，狐狸妹妹就把下载的东西存在了主人的家目录下的“下载”目录。这个所谓的家目录，就是/home 目录下以主人的用户名命名的那个目录。比如我主人的家目录就是/home/lanwoniu，如果有个用户叫做 kkk，那他的家目录就是/hom/kkk。我们 Linux 喜欢简单明了，每个用户就只能在自己的这个家目录下所以操作，其他的目录别动。不光用户的文档图片之类，包括这个用户所用软件的配置文件，也都在他的家目录下。主人找到了 deb 文件，双击。根据之前的约定，看到主人双击 deb 文件，我赶紧叫醒新立得来处理。新力得自然只是负责跟主人交流，还得叫醒超级牛力来干活。超级牛力拉过这个 deb 包，按照流程，先看包上面的依赖关系说明。一般上面会写着，要想装上这个包里面的东西，需要先安装 xxxx 库，xxx 软件之类的。超级牛力得根据上面的需求，对比自己小本子上关于本机已安装软件的记录，如果发现有什么没有装的，就自动去网上下载并且装好，当所有依赖关系都满足了，就打开这个软件包开始安装。Chrome 这个包的所有依赖关系都是满足的，于是超级牛力按照规范解开了这个 DEB 包，从里面拎出一只浏览器。

这家伙长的色彩斑斓，红黄绿蓝四色相间，样子简简单单，倒也清爽。大家没来得及仔细看，这家伙就被超级牛力带进硬盘里安排住处去了。狐狸妹妹有些不高兴的说：哼，回头倒要看看他有什么本事。 是啊，这家伙来了自然就会抢狐狸的饭碗，难怪狐狸不高兴。

没过多一会儿，Chrome 就被主人叫起来干活了。Chrome 一跑进内存，一下子分出4个进程，在工作间里窜来窜去，这叫一个热闹。4个 chrome 分别忙着各自的工作，有的负责向图形界面申请窗口，有的负责在窗口上绘制各种标签，按钮。有的打开网口，访问默认的主页，虽然这么多进程挺闹腾，倒也是各司其职。而且这样一来，这些事情是同时进行的，使得用户会感觉他快一点。相比之下狐狸妹妹就比较本分，一步一步进行。先向图形部门申请绘制窗口，窗口批下来之后再在窗口里绘制好各种文字，菜单，按钮之类的。这之后再根据用户设置的主页，上网申请数据，数据来了再显示给用户。有插件的话，在这之前还要加载各种插件，难怪有人觉得狐狸有点慢呢。这还不算完，等到主人真的开始是用 Chrome 浏览网页我才知道，原来这 Chrome 不只是4个，随着主人打开的网页越来越多，Chrome 不断的复制自己，工作间里的 Chrome 进程也越来越多。好家伙，转眼间工作间里就看不见别人了，到处都是 Chrome。不过好在他们每个占用的资源都不多，因此也不会给别人带来什么麻烦，这也算是 Chrome 这家伙的特色之一吧。

Chrome 来了以后，这工作间里面就开始不太平了。狐狸妹妹和他是谁也不服谁，整天吵架。系统中可以有很多浏览器，但是默认的浏览器只有一个，原本是狐狸妹妹。可当 Chrome 启动的时候就会告诉主人：我不是默认浏览器，把我设成默认浏览器吧。如果主人把 Chrome 设成默认，狐狸妹妹启动时也会向主人抱怨：以前我是默认的，干的好好的怎么就换了呢？您还是给我设回来吧。也搭上我们这主人没准主意，墙头草，随风倒。今天设狐狸为默认，明天改 Chrome 是默认。不过最终还是确定了狐狸妹妹的默认浏览器地位，毕竟是我的原班人马嘛。别的不说，由于狐狸是软件源里的软件，所以如果有个什么版本升级 BUG 修复之类的，都是自动的。那时超级牛力会接到 Canonical 学校发来的通知：狐狸同志，工作一向兢兢业业，刻苦提高技术水平，今经组织决定，晋升狐狸为 x.xx 版本，建议进行升级。得到主人的同以后，超级牛力就去下载来新的部件，把我们机器里的狐狸按照学校提供的步骤该装成新的狐狸。每一个我带来的软件和通过超级牛力安装的软件都有这种待遇，而像 Chrome 这种从网站上下载来安装的软件就不行了，要想升级，那就只能再去网上下载新的版本，自己重新安装。 所以 Chrome 们整天不服气，跟狐狸说：“你也就是因为有靠山，那笨兔子内核跟你一伙的，所以才能容得你这么飞扬跋扈。”狐狸自然也肯罢休：“你那样子又有什么好了，再说了，要变成你那样子简单的很。给我装上 Chromifox Basic 主题，再装上 Total Rechrome 扩展倒置标签兰和地址栏，最后装个 Hid Menubar 扩展隐藏掉菜单栏，跟你也没什么区别。只不过主人不这么打扮我罢了。”另一只 Chrome 反唇相讥：“哼，变成我们这样有什么用，装那么多插件扩展的，臃肿阿，哪像我们简洁高效。”狐狸怒到：“你想装这么多插件还没人给你开发呢！”……

整天这么闹腾也不是个事，听人家说养金龙鱼的话，一定要养单数条。因为如果一起养了两条，这两条就会争斗不休，直到两败俱伤为止。如果养三条就太平无事，因为任何两条鱼争斗，都不免让第三条鱼“渔翁得利”，因此三条反而太平。大概主人也直到这里道理，所以，他又装了一个浏览器——Opera。

Opera 小姐来自挪威，身材比狐狸妹妹好些，体态轻盈，长相倒是一般，总爱穿着一身大红的衣服走来走去，举手投足间透露出一种高贵的气质。Opera 最大的特点就是速度快，干活麻利。作为一个浏览器，最基本的技能就是渲染网页嘛，所谓渲染网页就是跟据从网站上取回来的 html 语言的描述，把相应的文字和图片以及 Flash 的元素显示

在页面上。所以浏览器每天的主要工作就是解释 html，以及 jsp 等等语言，并显示出漂亮的网页。Opera 在解释语言上有她自己的一套办法，解释起来迅速快捷。具体为什么快，没人知道，因为她虽然也免费，但却不像狐狸妹妹和 Chrome 一样开源。Opera 是一个闭源的软件，我们不能够了解她的内部构造。除了速度以外，Opera 也有扩展和插件，但由于她不开源嘛，她的扩展都是由她的东家——Opera 软件公司设计的，不像狐狸妹妹的扩展都是广大热心的网友们写的，所以 Opera 的插件没有那么多。另外，Opera 还是个多才多艺的浏览器，除了能做浏览器，她还能 irc 聊天，能作为邮件客户端，现在她又学了个新本事，叫 Unite，能建立简单的 http 服务器用来跟别人共享文件了。虽然狐狸妹妹通过插件，也可以实现一些功能，但毕竟是跟 Opera 差了那么一点。

这一来，三个浏览器成三足鼎立之势，任何两个浏览器都可能联合起来对付地三个。Chrome 发彪的时候狐狸妹妹会和 Opera 姐姐联合，因为 Chrome 一分就好多个，人多势众，而狐狸和 Opera 都是单进程的，团结起来才是办法；狐狸要犯脾气的话 Chrome 和 Opera 会联手，因为狐狸是我的嫡系软件，安装时就在还能自动升级的，跟其他软件的关系密切，Chrome 和 Opera 则都是外来的，手动安装的，初来乍到还人生地不熟，自然要结成团体；Opera 使小姐性子的时候狐狸和 Chrome 肯定站在一边，因为这俩都是开源的软件，大家的思想和理念比较一致，Opera 的闭源思维难免和他们会有格格不入的地方。于是三个浏览器互相牵制，工作间里面从此就天下太平了。

后来，主人又先后安装了很多浏览器，有极其简约的 Uzbl，有轻巧的 Epiphany，有简陋的 dillo，甚至纯字符界面的 lynx 和 links 都装上了。主人装这么些个浏览器干啥？一块运行起来看热闹玩？当然不是，主人是在体验，是在选择，寻找最适合自己的那一款浏览器。我们 Linux 世界里的软件总是多种多样，同样功能的软件可能会有很多款，而且特色各异。每个人都可以找到适合自己的那一款。甚至如果没有哪款和你心意的，你还可以通过修改某个浏览器的代码（因为多是开源的嘛），来创造出你自己喜欢的浏览器来。比如狐狸妹妹深受广大用户喜爱，然而就是干活速度有些满，于是有人就在狐狸代码的基础上加以改进，出现了疯狐狸（MadFox）浏览器。这就是 Linux 的世界，在这个世界里，人类才是软件的主人。

虽然主人有了狐狸妹妹的 downthemall 插件，但是毕竟功能有限，网络上很多的资源都是需要 bt 下载的，于是主人在选择了适合自己的浏览器之后，又开始了 bt 软件的海选。

主人的海选工作主要通过网络进行，由狐狸妹妹和狗狗哥作为星探，到世界各地打探变态人才（变态，也就是 BT 啦），并到主人处报名登记。之后经过主人的初步筛选，由超级牛力负责召集获得参赛资格的软件前来比赛，超级牛力力所不及的（也就是不在源里的啦），就由狐狸妹妹去邀请。把变态软件们找来之后，进行初赛，由评委投票表决，评委们只能表决“通过”或者“否决”（二进制嘛）。初赛之后，将结果交给主人，由主人决定选用哪几个软件，进入复赛。

好，现在诸位选手已经来到了硬盘里（也就仗着我们这硬盘地方大），初赛马上开始。在初赛中，每位选手首先要展示一下自己的外貌，之后下载一首固定曲目并汇报下载速度。评委根据选手的外形，速度，占用情况进行评判。有两位以上评委判选手通过即可进入复赛。哦对，忘了介绍了。本届变态软件大赛评委一共有三位，首先是 ifconfig，大家欢迎。（哗哗……掌声）ifconfig 是目前我们这里的网络专家，主要关注选手的下载

能力。第二位是 Top，中文名叫“顶”。（哗哗……掌声）你看论坛里老发帖子说“顶”，那就是说他呢。Top 老师专注于选手的资源使用情况。第三位是 X，图形部门的老大，他会评判选手的外形相貌。当然，最终的总评委，那还是我们主人啦。好，现在比赛正式开始，首先请第一位选手出场。

"大家好呱，我叫 Vuze 呱，大家可能对我的新名字不大呱熟悉，我之前呱叫做 Azureus，中文呱名字叫做毒蛙，呱呱。","打断一下阿"，top 说道,"我看你的样子，你是 java 的吧？""恩，对呱，我是 java 的软件。""哦，好了，开始吧。" "好，下面我为大家演唱，哦不是，我为大家下载呱。"他一边说着呱——咳，我怎么也呱上了-_-b。他一边说着，一边拿起了固定曲目的种子文件——就是那种.torrent 的文件，然后开始了下载。X 评委看到 Vuze 选手相貌出众，仪表堂堂，下载的时候也是有条不紊，进度，速度，显示的清清楚楚，并且效果也算不错，不禁心中对他就多了几分好感。这时，Vuze 已经连接到了服务器，找到了下载源，也就是找到了种子，开始下载了，不过找到的种子不多，速度也稍微有些慢。ifconfig 评委觉得有些差强人意，不过也算合格。Top 评委则一直在下面叨咕着：“这漂亮倒是漂亮，这资源可也不少占阿。他这么一上台，这200多 M 的内存可就没了。这 java 的程序真是占地方”一边叨咕一边轻轻的摇头。等到 Vuze 下载速度稳定之后，评委叫他暂停说：“好，可以了，下一个吧。”最终 Vuze 的结果是 X, ifconfig 两评委通过，top 否决。

接下来是第二位选手，Deluge。

“Hello，大家好，我是 Deluge，今天很高兴能够在这里与各位老师交流，我想不管结局如何，我都会有所收获。我还比较年轻，大家可能对我的名字不如前面的 Azureus 同学那样熟悉，不过我也已经在 Windows，linux，MacOS 都有所成就” “哦，这些系统你都接触过”ifconfig 问。 “是的。” top 扭头对另外两位说：“刚才忘了问 vuze 能不能跨平台了。不过这也不是主要的。” X 不懈的说道：“人家是 java 的，自然能夸平台。”之后，向 deluge 点点头说：“恩，你可以开始了。”只见 deluge 拿起刚才 vuze 的那个 torrent 文件，开始连接服务器，找种子，下载。找种子的速度比 vuze 快了一些，下载起来速度也比较稳定。ifconfig 说道：“你看，他就比 vuze 下载能力强，速度稳定，刚才 vuze 是忽快忽慢，估计是因为 vuze 找到的种子少。”top 接着说：“恩，身体也灵巧，才占了不到30m 的空间，而且 cpu 用的也少，这个不错：” X 淡淡的说：“样子倒是中规中矩，说不上好看吧，倒也简洁清爽。”最终 deluge 全票通过。

“老师们好，我是三号选手 qbittorrent。”刚说一句话，只听 X 在台下小声说道：“哼，K 派的，到咱这来干活还不知道会不会整天吵架呢。”台上倒是并没有受他影响，继续说道：“我出生在法国，不远万里来到这里，把这里的变态事业当做是自己的变态事业，这是什么精神？我觉得这就是国际变态主义精神，这就是……” 下边 top 受不了了：“行了行了，赶紧开始吧，这段我们都背过。”这 qbittorrent 赶紧开始，很快找到种子并且下载上了。ifconfig 一看：“这小子不错哈，这速度比那个 deluge 又快了不少，得有1.5 倍了，呀，又涨了，奔着两倍去了。”正说着呢，qbittorrent 一个没留神，把种子给丢了，速度瞬间到0，重新找种子。ifconfig 擦了擦头上的汗说：“就当我什么也没说吧。”之后直接选了否决。X 却微微点头，看着旁边的 top 说：“还可以吧？至少我看样子还可以，倒是有 K 派的风范。”top 也表示认可：“恩，这个占用的资源又比那个 deluge 少了，难得难得。”qbittorrent 获得两票通过。

qbittorrent 下去后，又上来一个，自报家门：“偶系 ktorrent，偶超喜爱 B 态协议，喜爱做 B 态软件，因为冷够让大家婚享快乐滴讯间，偶能够做一个酱紫滴软件，偶好

高兴好高兴。”ifconfig 指着 ktorrent 问身边的 top：“这……这90后吧。”top 无奈道：“废话，这些软件哪个不是90年以后的，90年那会连 Linux 还没有呢。”X 低着头看着自己的本子向 ktorrent 摆手：“行了行了，开始吧。”于是 ktorrent 拿起那个 torrent 文件，开始干活，首先找种子，找了5分钟，没找到。ifconfig 不耐烦了：“我说……咱国庆节前还能找到不能？”ktorrent 很谦虚：“哦，元旦以前，一定可以找到滴～”要说还是 top 办事果断，喊：“下一个！”全票否决。

过了一会，听见下一位选手说：“我呢，叫做 bittorrent，听我这名字就知道……”还没说完，三位评委同时惊讶的抬起头，看着空荡荡的台子问道：“你在哪呢？”只听那个选手说：“我在这呢，我在这呢，你们站起来就看见了。”仨评委站起来一看，好家伙，这哥们还真够小的，刚才让桌子挡上了，看不见。只听那选手继续说道：“别看我小，我可是符合 linux 哲学之美的软件，短小精干，我小是因为我没有图形界面，咱就专注于干活，专注于下载。下的快，占用资源还少，这比什么都重要。”top 听了微笑着点点头，直接选择了通过。X 说：“虽然这样，但是目前看主人的意思还是应该找一个有图形界面的，不是你能力不行，是这里的工作有些不适合你”之后选择了否决。现在就看 ifconfig 了，ifconfig 向来主张实力压倒一切，所以让 bittorrent 下载一下看看速度。然而 bittorrent 也没能表现出比较优秀的速度来，还一个劲解释：“这个，你们这里的网络的问题，你们这是内网，所以……”ifconfig 没心思听他解释，终于选择了否决。

最后一位选手还没出场，就听见他的高声朗诵：“君不见，黄河之水天上来，奔流到海不复回？君不见，高堂明镜悲白发，朝如青丝暮成雪。”念完这句，他也从后台走到了前台。只见他一派仙风傲骨，飘逸洒脱。top 问到：“你叫什么名字呢？”他说：“我的名字，便在这首诗文中。”X 说：“你叫黄河？”他答道：“我叫奔流。”top 问：“听你的朗诵，你来自中国？”奔流答到：“是的。”ifconfig 说：“看看你有什么本事吧。”奔流二话不说，拿起 torrent 文件就开始忙活，很快找到了种子，下载速度也一路飙升，三位评委都非常满意。奔流最终全票通过，不过评委们在最终的结果上做了一个标注：此软件不开源。

初赛结果递交给主人，主人看了之后，决定两个获得全票通过的软件来进行复赛：deluge 和 奔流。复赛的裁判只有一位，就是主人自己；复赛的原则只有一条，就是用着顺手；复赛的时间，是一周。

主人用了一周后，总结了一下这两位软件各自的特点。下载的功能上来说，两位选手都是不错，不过奔流依然是在速度上领先一步，可能因为他生在中国，对这里的网络状况更加了解吧。其他周边功能上，由于 deluge 支持插件，所以能够扩展出一些有用的功能，比如 Web User Interface 插件，可以让用户通过网络来管理这台计算机上的 deluge 下载。如果你的电脑直接链在公网上的话，你甚至可以在单位通过网络查看家里电脑上的 deluge 的下载任务。除此之外，由于 deluge 是源里的软件，能够自动跟着系统一起升级，也算是一个优点。然而，主人还是不大关注这些附加的功能，主张速度优先，既然奔流更快，就选择了奔流。最终胜出的奔流非常得意，每次下载的时候，找到一个种子，奔流都会炫耀的大声喊“找到一个种子！”“又找到一个种子，哈哈”。当然，主人也很烦，把他的这个功能关了。deluge 呢，也没有被删除（其他的那些选手都被删除了），作为二线部队备用。

故事外的事——位数

上面提到了，我是个64位的操作系统。这个多少多少位，说的是 cpu 一次运算的二进制数字的位数。这个 CPU 就像是个计算器，我们软件用 CPU 就像人类用计算器似的。它很重要，我们要算一丁点东西，也需要用 CPU 来算。（别跟我说用心算，我是软件，ok?）那么这个 CPU 算东西的能力，是有限制的，有什么限制呢？你拿出你家的计算器看看，算个28+783，没问题是吧。算个7836-473也没问题是吧，再算个726354465849392029373465 37+1，能么？估计99%的同志出问题了(不排除有牛人拥有很牛的计算器)："我哪能按出这么多数来啊，我这计算器总共就能显示下11位数字"。对，这就是计算器的位数限制。CPU 也一样，他一次能算的数不能无限的大，总得有个边，只不过不是按照十进制的位数算的，而是按照二进制的位数算的。至于什么叫十进制，什么叫二进制，可以去问问狗狗大哥，不过不知道也没关系，咱暂时按着咱们平常的十进制来说。我们软件是用 CPU 运算的过程和你们人类是用计算器是差不多的。比如说，我这有个计算器只能算99以内的数字，也就是只有2位（也不知道谁设计的这么弱智的计算器）。那么我用这个计算机算个3＋4怎么算呢？简单，输入3，按＋号，再输入4，按＝号，就出来了。再算个大点的，算个56＋47。先输入56，按＋号，再输入47，按＝号。疑？显示03，怎么不显示103呢？废话，它倒是想显示，往哪写那1呀？但是我用的这个计算器（也就是 CPU 阿）是很人性的，会提示你运算结果超出了它的能力范围。比如可能会有个红灯亮起，提示你03前面还有一个进位，进到百位了。

好了，基本的操作说完了，现在说正题，不同位数的区别。两位的 CPU 就像刚才说的那样，那么假设现在需要计算3173+644，这里有2位的 CPU 一个，4位的 CPU 一个，分别用他们做这个计算，有什么区别呢？

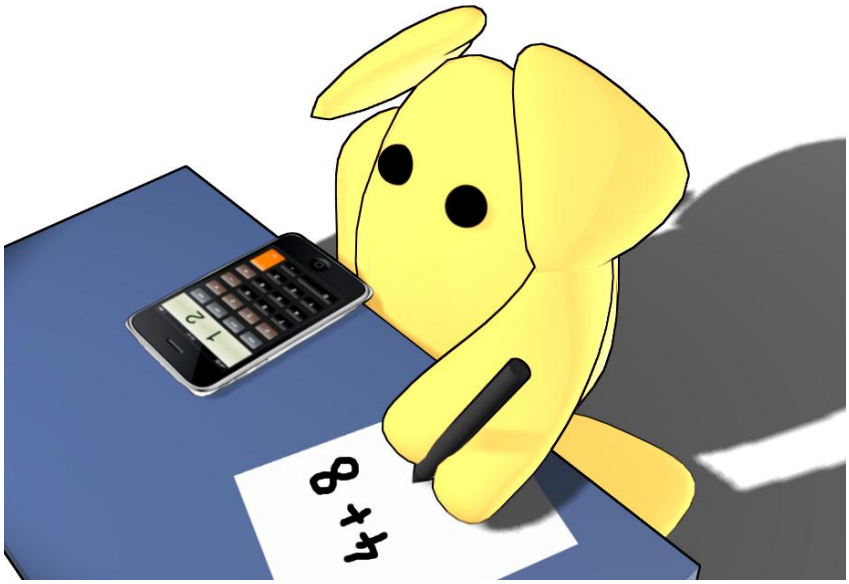
咱先拿这两位的，有人说了，两位的只能算两位啊，这个没法算哪？唉，这机器是死的，咱软件是活的啊，一次只能算两位，咱不会分开了多算几次么。首先，输入73，按+号，再输入44，按=号。显示出来17，同时红灯一亮，说明还得进位。好，找张纸记下17这个数，还得写上"得进位"。然后再输入31，按+号，输入6，按=号，显示出来37。别忙，没完，刚才还得进位呢么不是，再输入37，按＋号，输入1，按钮，咔嚓，出来38。好，最后结果拼一块，高位是38，低位是17，最后结果：3817

再拿这4位的算算看。4位的就意味着输入的和显示的数最大可以是9999，也就是说我直接就可以输入3173，按+号，再输入644，按=号，显示出来3817，ok，收工～

这就是2位的 CPU 和4位的 CPU 的不同，从理论上来说，4位的要比2位的快，从上面的例子看的很明显嘛，大一点的数，4位的 CPU 一下就能算完，2位的 CPU 要折腾好几次。但是这4位的 CPU 还得有人会用才行，这就需要4位的软件来用着个4位的 CPU。

终于说到软件的位数了，CPU 的位数就是一次能计算多少位的数，那软件的位数呢？就是说明这个软件需要使用多少位的 CPU。软件干活肯定需要计算，计算就得用 CPU，2位的软件会用2位的 CPU，4位的软件就会用4位的 CPU（还是拿十进制位做比喻啊）。比如有一个2位的软件（就说明这个软件会用2位的 CPU），那么当这个软件运

行在一个2位 CPU 的电脑上的时候就是这样:还比如要算3173+644，他就会先算73+44，然后记住进位，然后计算31+6，然后加上进位，最后拼起来，得到答案，就像上面描述的那样。那么当这个2位的软件运行在一个4位的 CPU 上的时候会怎么样呢？他会先算73+44，然后记住进位，然后计算31+6，然后加上进位，最后拼起来，得到答案……有人说了，他怎么不直接算啊？4位的 CPU 不是能直接就算出来么？但是别忘了他是两位的软件啊，他不会用4位的 CPU，但是不会用不等于不能用，他还是可以那4位的 CPU 当成2位的来用，只是有些浪费而已。那么要想完全发挥4位 CPU 的性能怎么办呢？当然就得4位的软件出场了。当一个4位的软件运行在一个4位的 CPU 上时怎么计算3173+644呢？大家大概都知道了，直接算，一次完成。那么当一个4位的软件运行在一个2位的 CPU 上时会怎么样呢？这个软件会写个3173的纸条要往 CPU 的寄存器里塞，急的满头大汗就是塞不进去，最后一甩手——老子不干了，这破 CPU 没法用！当然，这只是个比喻，并不是说4位软件在2位 CPU 上算3173+644就算不了，算1+1就能算。4位的软件是根本无法运行在2位的 CPU 上的。



故事外的事——进程

咱说 Chrome 是个多进程的浏览器，一运行就复制出好多进程来。有人可能对进程这个名字还不是很明白，什么是进程呢？简单地说，进程就是正在干活的软件。比如狐狸妹妹,躺在硬盘里睡觉的时候他就是一个软件,一堆数 据，一陀代码。当他被叫醒，跑进内存里开始干活的时候，他就是一个进程了。（当然，其实这么说不很准确）换句话说，内存里忙忙碌碌的，都是一个个的进程。 当然，同时他们都是程序，都是软件，这个不冲突。就像去公司上班的人，他们都是人，当他们在公司工作的时候，他们都是公司的 员工。员工，就像进程一样。很多公司的员工每个人都有个工号，什么

NB001,SB999之类的，每个进程也都有一个唯一的标识——进程 ID 号，简称 PID。这个 ID 号是由我分配给每一个跑进工作间的进程的，分配的规则很简单，每人一个，每次加一。第一个跑进来的就是1号，上面介绍过，Init 这家伙 每次都是第一个被我叫起来，帮我打理一下日常工作，所以他的 ID 号总是1。而且，他还有个特殊身份，这个呢，咱暂时保密，待会再说。

每个公司的员工都有个直属的上级，上级又有上级，以此类推。我们这里的进程也是这样，只不过我们不叫"上级"或者"上司"，我们叫——爹！好吧，似乎这个 称谓土了点，但是意思就是这个意思。一个进程之所以成为一个进程，一定是由于另一个进程创建了他。（有点绕嘴吧）比如说主人来了一个终端，于是就有了一个 bash 进程，然后主人在这个终端里敲入 firefox 然后回车，bash 就知道这是要他去叫狐狸妹妹来干活，于是 bash 就去找狐狸妹妹，把她带到内存 里开始工作，于是就创建了一个 firefox 进程。好了，现在，firefox 这个进程是由 bash 这个进程创建的，那么，bash 这个进程就是 firefox 这个进程的父进程，firefox 进程就是 bash 进程的子进程，也就是说，狐狸妹妹就得管 bash 叫爹！那 bash 也得有个"爹"吧？是 的，如果是在 Gnome 环境下开的那个终端的话，那么 bash 它爹就是调用 bash 的 gnome-terminal。那么如此循环往复，肯定有一个站在金 字塔最高点的总"爹"吧？难道，难道笨兔兔你就是他们的总爹？很遗憾，我不是，所有进程的总爹，是每次启动第一个被我叫起来的 Init，所有的进程都是被 init 直接或者间接创建的，Init 才是所有进程的祖宗！

关于父进程，有两点要说明：

第一，我们这的父子关系不是固定的，是会变换的。如果从 bash 启动 firefox 那 bash 就是 firefox 的爹，如果直接从图形界面启动那就没 bash 什么事情了。（这时候 firefox 的爹其实是 init）；

第二，不要问我哪里有妈进程！

当爹也有当爹的义务，人家不能白叫你一声爹是不是。 当自己的娃（也就是子进程啦）做完自己该做的工作以后，就停止了一切动作，像个死尸一样待在那里，当爹的就负责给他"收尸"-_-b 一个结束了所有工作的进程，会处于一种"僵尸"状态，这时候他什么也不做了，就等着被干掉。进程进入僵尸状态前一般会通知他爹一声，汇报一下说：爹啊，俺 已经把该做的都做啦，现在我要变僵尸啦！（让后平举双手开始行走？那是生化危机！）然后他爹负责向我汇报：我家娃干完活了，你把他的工号（就是 PID，记得吧）取消掉然后让他回去睡觉吧。然后我就把它的工号收回，然后看看他有没有什么申请了没释放的资源（一般一个好孩子在结束运行成为僵尸之前会主动释放掉 自己申请的资源的。），确认都没问题了之后，他就被从我的进程列表中清除了。但是有时候也会有些特殊情况，比如有的时候娃还在兢兢业业的干活呢，结果他爹 死了。（可能他爹干完活退出了，也可能被主人用命令 kill 了。）这个时候我就会发个信号给他家娃说：那个……娃呀，那啥，跟你说个事，你爹死了。这时候 有的娃就悲痛欲绝：俺爹都死了俺活着还有啥意思啊，呜呜呜～～～俺也僵尸吧。然后就退出了。比如你在终端运行 firefox，然后把终端关 了，firefox 也就退出了。也有的娃比较坚强，一定要完成上级交给的任务，化悲痛为力量，这时候我会给他找个新爹——因为每个进程总得有个父进程，没 爹是不行的。一般我会安排他爹的爹来当他的爹（又绕进去了吧），也就是这个进程原来的"爷爷"进程来当他的父进程。然后这娃在长了一辈后，继续认真工作。 比

如你在终端运行 nohup firefox，然后把终端关了，firefox 继续运行。那如果他爷爷不幸也挂了呢？那就继续往上导吧，我们说了 Init 是所有进程的祖宗，所以他那里 成了最终的“无依靠青年进程收容所”。

还有的时候娃已经把该做的事情做完了，汇报给他爹并变成僵尸。可是他爹还没来得及给自己娃收尸，自己就挂掉了。这个时候，我没法通知那娃说她爹挂了，因为 那娃已经是僵尸了，啥也不听啥也不干了。可我也不能直接把他干掉，啥事情都得按规矩来嘛，只有他爹向我申请我才能把他干掉，可是他爹又已经挂了……那怎么 办呢？那就按流程来，先给这个娃找个爹，哪怕这娃已经是僵尸了，也得有个爹。比如我找到 init 说：那个 ID 号是2725的那个进程爹死了，你当他爹吧。 一边说一边看也不看的用手往那边一指，假装自己没看到那娃已经成僵尸了。 一般 Init 也不会太注意，直接就答应了，然后马上发现了事情的真相，跑到我这里来说：那娃已经成了僵尸啦，你还叫我收养个啥？我肯定会一脸无辜装：啊？ 是啊，那不管怎样，你是他爹了，你负责处理一下后事吧。于是 init 只好以爹的身份处理那个僵尸的后事，问题就这样解决了。

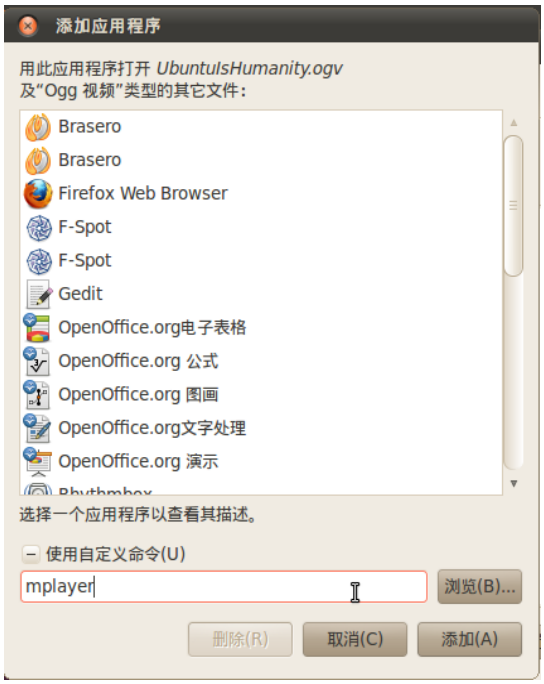
3.3 我的影音生活

狐狸妹妹今天接到主人的任务，要去下一部叫做《Big Buck Bunny》的电影，这部电影长达9分56秒——还没电视节目中间的广告长，但是他有一个特点，一个和我一样的特点——他是开源的。电影怎么个开源？他是在开源的平 台上用开源的软件制作，并且免费下载观看还可以获得他的原始制作文件（blender 的文件），如果你愿意，还可以进行修改，编个续集什么的。好，废话不多说，这个电影提供了 http 下载，于是这件事情自然由狐狸妹妹拿着她的 Downthemall 扩展来搞定了。

下载下来之后，主人就找来播放器来播放这个视频。之前主人安装了 Gnome Mplayer 播放器，这个播放器其实只是个图形前端，真正在后台默默无闻的播放视频的是 Mplayer。用过一段时间之后，主人觉得这 Gnome Mplayer 的界面也挺简陋，而且看片的时候其实基本不需要怎么操作，那还不如干脆不要这界面，直接让 Mplayer 来播放，就光一个视频窗口其他的什么按钮都没有那才简约，而且也很拉风，因为在 windows 下没有这么看视频的。想到这里，说干就干。主人找到了下载好的那个 ogg 文件，这是一种开源的视频文件格式，mplayer 不用安装额外的解码器就可以播放的。主人用右键点击了这个文件，选择“属性”。

找到"打开方式"标签，上面显示了目前在我这里注册过的两个软件可以用来打开这类型的文件，一个是 Gnome Mplayer，一个是电影播放机，也就是 Totem。并且电影播放机是默认程序，也就是双击一个.ogg 文件，就会用电影播放机打开——这些都不是主人想要的。

主人想用 mplayer 来打开这个文件，但是这里没有，那自然是点击了“添加”，然后我给他列出了一个软件列表，都是已经安装了的软件，如果是其他软件，在这里找就好了。不过 mplayer 是个命令行的软件，并不在这个列表中。但是没事，主人很熟练的展开了下面的那个"是用自定义命令"，然后输入了 mplayer，再"添加"。



这样，就回到了"打开方式"那个标签，上面已经多出了一个没有图标的 mplayer，但还不是默认，再把他选成默认的程序，就好了，直接关闭。这样做了之后，再双击那个刚刚下载的 ogg 文件，mplayer 就自动跑出来播放了。只见屏幕上只有一个简约的不能再简约窗口，窗口里面就是正在播放的视频——一只可爱的大兔子。

有人说了，这界面上啥也没有，我想暂停，快进，调整音量啥的怎么办？不用担心，mplayer 早为你想好了，整个键盘就是你的操作界面。p 建就是暂停，再按任意键就是播放。按方向键“右”是向前快进10秒，"左"则是后退10秒。"上"快进一分钟，"下"后退一分钟，"PageUp"，快进10分钟，"PageDown"后退10分钟。这样，打开一个视频后就可以方便而精确的找到想要看的段落。你现知道现在播放到了多少分钟？按"o"键试试？调音量，"0"键升高，"9"键降低。遇到音画不同步的情况，用"+""-"号调整音频的延迟时间。如果是外挂字幕的，还可以用"z","x"调整字幕的延迟时间。还有"1","2"调对比度，"3","4"调亮度，"5","6"调色相，等等。主人早已熟悉了这些快捷键，现在操作起来真是得心应手。

主人是个喜欢简约的人，他很有成就感的看到自己把一个作系统的播放器整的这么精简，不过也不是所有人都受得了只有一个窗口的播放器的。

在这一天，一个陌生的，温柔的手指触摸到了我们这台电脑的键盘，接着，一个清馨的声音透过一直插在电脑上的麦克风，回荡在紧张忙碌的工作间里：“呀，你的电脑真好看，你这是 win7吧？”哎，一听就是个外行，win7哪有我漂亮，哼。随即听主人说

到：“这个不是 windows，是 Linux。”工作间里的同志们会心的点点头。

"哦？Linux？是微软新出的嘛？" 靠，除了微软不认识别的公司了阿。

"不是，这个跟微软没关系，Linux 是一个自由的操作系统，免费的哦"

"免费的？那能好用么？"姐姐，便宜也有好货哦。

"你试试阿，挺好用的，至少界面就好看"

"那倒是，嘻嘻。"

"其实你可以装一个看看，反正你不怎么玩游戏。看个片阿，上个网阿啥的很方便的"

"对了，你这里有什么有意思的电影么？"

"有阿，有个动画片挺有意思的，这个"主人说着，双击了那个.ogg 文件，mplayer 出现了。由于省去了运行图形前端（如 gnome mplayer），再由图形前端调用 mplayer 的过程，mplayer 出现的很迅速，光秃秃的样子也很让那个 mm 惊讶。

"这是什么播放器阿？"

"Linux 下的最好用的播放器，怎么样，简洁吧？"主人很得意。

"真难看～"得，碰钉子上了。

"这样什么操作界面都没有，难道每次必须从头看？也不能停？"MM 继续说着。

"当然不是"，主人解释："都可以用键盘操作的，你看这个是播放，这个是快进，这个调节音量……"主人一边解释，一边在键盘上演示着。

"还要记住那么多键阿，好麻烦"哎，艺术本来也不是每个人都能欣赏的。

"也有普通的图形界面的播放器，比如这个"主人关掉了光秃秃的 mplayer，打开了 totem。

"恩，这个还顺眼点，不过好像功能也不强大哦，有没有暴风影音？"

"要暴风影音那样的播放器也有阿，等我装一个你看看"说着主人打开了新立得，输入了 smplayer，然后选择了安装。数分钟之后，装好了，超级牛力干活没得说。

"这就装好了，怎么没见到你上网下载阿？"

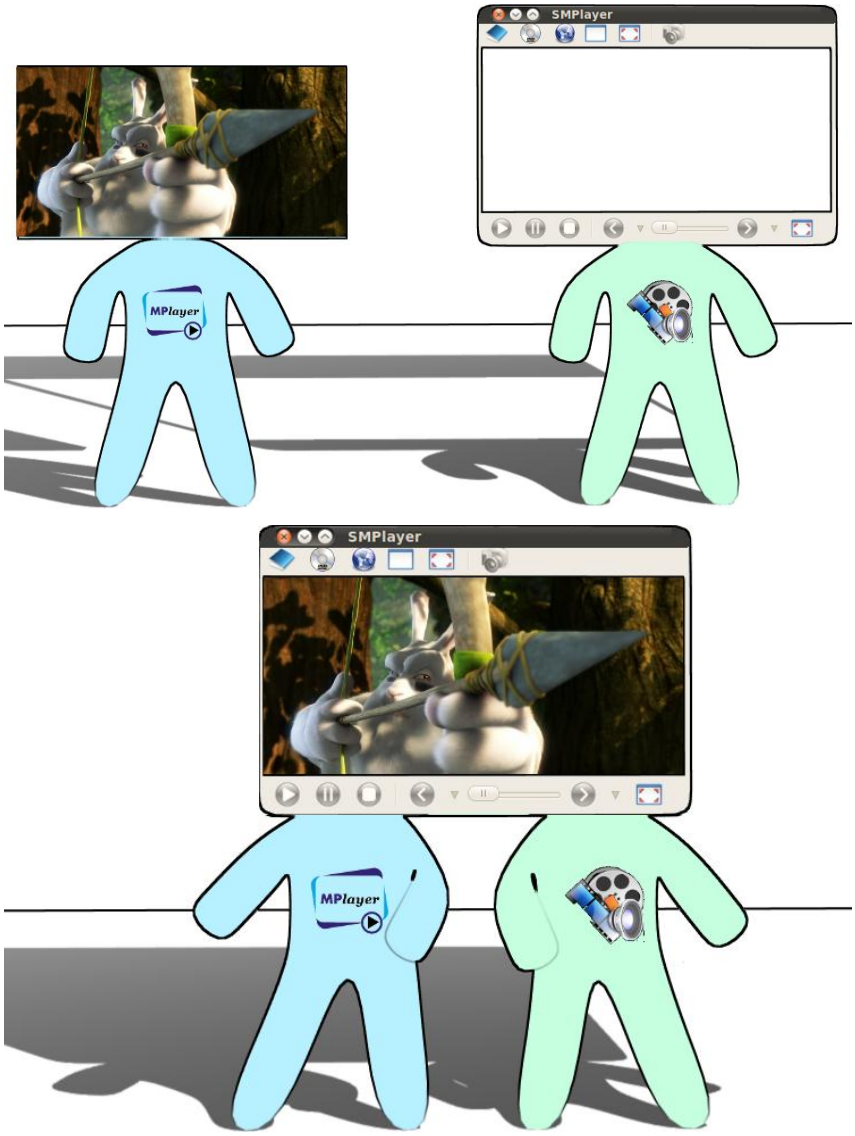
"这个……Linux 就是这样的。"我理解主人，要给这家伙讲清楚有点难。

"哦，这还挺方便。咦，这个界面好多了嘛，你之前怎么不用这个？"

"那个精简，速度快。linux 的软件就是这样，什么样子的都有，总有一款适合你，哈哈"

说着笑着，那个大兔子片就结束了，主人又让 smplayer 去播放一个叫做见过大爷.rmvb 的片子，并且和 mm 安静的看起来。Smplayer 其实跟 Gnome Mplayer 一样，都只是图形前端，还是离不开 Mplayer。但是 Smplayer 的厉害之处在于他能够充分发挥 Mplayer 的功能，提供更多可配置的选项，并且操作起来也更符合主人的习惯。可以说，

Smplayer 和 Mplayer 合作起来，那真是珠联璧合，黄金搭档。



主人和 mm 看着片子倒是安静了，殊不知工作间里面这会可闹腾起来了……

刚才主人用 Mplayer 播放视频，后来换成了 Totem，最后又改成了 Smplayer。这么折腾，Mplayer 老成持重倒是没什么，那被替换下来的 Totem 显然很不服气，抱怨说：“要论真本事，我也未必比谁差，不就是长的漂亮点么，哼！”这 SMPlayer 听不过去了，一边忙着播放视频，一边反驳：“你本领不差，难道说我本领差了？你能记住每个视频上次播放到什么位置么？我可不光是靠长的漂亮。不过话说回来，我们 Qt 的程序，倒也确实比你们 gtk 的细致了不少，用户爱看我们，这也是没办法的事情。”Totem 怒目而视：“你别忘了，现在内存里面的可是我们 Gnome 环境，你敢在这说这种话，有点欺人太甚吧！”“Gnome 怎么啦，Gnome 就是不如 KDE 好看，有错么？”我一听，坏了，这俩又要打起来。

话说在 Linux 这片恩怨情仇的大地上，做图形界面这行当的，以两大帮派为主。一个是 Gnome 帮，帮中的软件们都修炼 gtk＋宝典，帮众们信奉简单高效的做软件准则。

什么事情，都力求用简单的界面来实现，并不留给用户太多可以设置的东西。G 帮认为，对于初级用户，不要搞那么多设置项，搞的用户头晕脑涨。能默认的都默认好了，打开软件就工作。而对于需要自定义的高级用户来说，直接去改配置文件就好了，这难不倒高级用户。另一大帮派，就是 KDE 派了，K 派们都修炼 qt 大法，派中的软件都觉得界面要做得方便，易用，易于配置，坚信细节决定成败。界面要细腻，要漂亮，让人家一看就喜欢，这才是好的图形界面软件。所以 K 派的软件都有好多可配置的选项，新手可以无视，老手配置起来也很方便。两帮派观念不同，本来也没有谁强谁弱，但是偏偏有时候还是会争个风头，动不动就吵个天翻地覆。本来我这里都是 G 帮的软件，结果今天主人偏偏安装了 K 派的 smplayer，这不就吵起来了。

Totem，那可是 Gnome 的嫡系软件，自带的播放器。本来被主人换下来，心里就不舒服，再一听 SMPlayer 这么说他，更是无名火起，顺手超起一把兵刃——一个视频文件，指着 SMplayer 说："有本事，别光耍嘴上的功夫，咱们过两招瞧瞧。"不等 SMplayer 答话，刷刷刷原地耍了起来，只见他一招一式，无甚惊人之处，却招招使的灵活熟练，那视频文件就好像本来写进他自身的代码段里一般，什么播放，暂停，前进，倒退，全屏，截图，调横纵比，显示字幕，虽说只是播放器的初级功夫，难得是样样做得恰到好处，不温不火。这时那 SMPlayer 正在给 lili 用户播放视频，本不该分神，却忍不住争强好胜之心，也找个兵器练起来。他一边给 Lili 播放，一边跟 Totem 比武，双手同时耍开两个视频，速度自然是慢了一步，但确是花样叠出。什么音画时差调整，什么字幕控制，网上查找字幕，降噪，反拉丝，旋转屏幕，专门找那 Totem 不会的招式操练。Totem 见对方同时耍着两个兵器，虽说稍稍缓慢一些，却依然稳扎稳打，何况招事确然比自己多，自觉是落了下风。可他又怎肯示弱，扔下手上这兵刃，又从兵器架上超起另一把，练了一会又换一个，顷刻间，已经换了18件兵刃。什么 rmvb,mov,avi,wmv,flv,cd 音轨,mp3,mp4等等等等，件件是拿的起，放的下。那意思，别看我会的招事不如你多，但我能耍的兵器可不少。SMPlayer 见到此景，微微一笑，扔了手中的家伙，把 Totem 换下的兵器一件件拿起来，依次也耍了一通，却也是样样精通。只看的 Totem 一身冷汗，自知自己本领也就到此为止了，可是却如何下的了这个台？正自思虑之间，只听得身背后有人言道："贤弟，你且下去歇息，哥哥我来会会他！"

只见 Totem 身后闪出一人，正是 GNOME Mplayer。一看这名字您就知道了，这是 G 帮的人。SMplayer 一看见他，顿时也是心里犯嘀咕。为什么呢？您众位大概有所不知，这 GNOME Mplayer 和 SMplayer 其实都不过是个图形界面的外壳而已，他们的后台都是 Mplayer 老大。因此上，这二人本是同门兄弟，只因信仰不同，故而一入 G 帮，一投 K 派。即是同门学艺的兄弟，这本领又能差的几何？这 SMplayer 寻思，纵然自己比 GNOME Player 勤学苦练，无奈方才刚刚跟 Totem 大战一场，有要照顾这那 lili 用户的视频，自己恐难取胜。可那 GNOME Player 更无废话，上来就超起兵刃比划起来，也是将一件件兵刃轮番耍起来，比刚才 totem 所耍的还要多上一些。SMplayer 只得咬咬牙，再跟 GNOME Player 拼死一战。只见整个内存中那真是录像与配音齐飞，字幕共长片一色。二人杀的是难解难分。这 GNOME Mplayer 平时便不服气 Smplayer，本是同门学艺，可在江湖上，绿林中，这 SMplayer 的名号却远远盖过他 GNOME Mplayer。今天他就想让别人见识见识，他 GNOME Mplayer 不比 Smplayer 差。因此他论起武艺，虽然与 Smplayer 差着一点，但是招招都拼尽全力，狠辣异常。斗到一百三十多回合的时候，这 SMplayer 渐渐支持不住，眼看就要跳帧了。这时候，内存里已然围满了人。有看热

闹的，有叫好助威的，有打酱油的，有路过的，还有不明真相的群众。围的是里三层外三层。虽然主人也安装了 KDE 环境，可是这次开机，进入的是 GNOME，因此这内存里围观的基本都是 G 帮的人，K 派的人都在那硬盘里睡觉呢，Smplayer 可谓是孤军奋战了。

正在这时，OpenOffice 老先生推推眼睛，咳嗽了一声，发话道：“两位请住手！”OO 老先生可谓是德高望众，虽然本身是 G 帮的人，可基本所有的 Linux 发行版，不管是用 GNOME 也好，用 KDE 也罢，都会请他做默认的办公软件。因此上即便是 K 派的人，也都服气这位老先生。话音刚落,GNOME Mplayer 罢手不打,SMplayer 自也愿意停下来休息,站在那里直喘粗气。只听 OO 老先生继续说道：“要我说阿，这机器启动时进入的是 GNOME，满内存都是 G 帮的人，可用户偏偏要叫来人家 SMPlayer 来播放视频，这已然是人家赢了。更何况两位轮番上阵，不是也没把人家一人打下来么？”后面这句，自然是看着 Totem 和 GNOME Mplayer 说的。Totem 艺不如人，自觉惭愧。GNOME Mplayer 愤愤不平，却也无话可说。一场风波暂时算是平息了。

内存里折腾的时候，主人和 MM 一边看着片子一边闲聊，倒是十分惬意。他们聊到生活，聊到感悟，聊到春花秋月，聊到诗词歌赋，还聊到一个挺有意思的片子《一头猪的故事》，据说思想性艺术性很强，尤其他的那首主题曲《猪之歌》，据 MM 说是感人至深，发人深省，于是 MM 走后，意犹未尽的主人去决定去找了来听听，聊以慰藉今日畅谈。

播放音乐这件事情，基本上 mplayer,totem 等视频播放器都可以完成，因为本来一个视频中就包含视频编码和音频编码，除非你看的是哑剧。一个视频播放器要同时能够解视频的编码和音频的编码，那么对于只有音频的文件来说，播放器来当然不在话下。这视频播放软件，就像武侠小说中那些飞来飞去，飞花落叶皆可伤人的侠客。他们有凡人无法想象的武功（其实都是想象出来的），S 级别的单挑能力。然而如果是领兵打仗，排兵布阵，就需要熟读兵书的将才。作为武将，马上的功夫也是少不了的，然而更重要的是能够更好地治理军队，指挥千万人作战。音频播放软件也是如此，论解码，他和视频软件都需要会，然而视频软件专注于怎么将视频解析的更加清晰，播放的更加流畅，音频软件要解的码量少，不是专业人士的话，对解出的音频质量也没有更多的要求，而更重要的是对文件的管理，和与上级（也就是用户啦）的交流。或者说，音频播放器，就像一个图书馆的管理员一样，要对自己的资源了如指掌。

我们这里，系统刚刚装好的时候就有专业的音乐管理员，就是跟随我一起到这台电脑来的 Rhythmbox。这哥们长相一般般，不过功能不差，一般的音乐自然都能摆平，网络电台啥的也没有问题，还有获取歌词和专辑封面（不过基本都是英语歌曲），管理起音乐文件来出看似乎有些凌乱，不过熟悉了之后就明白他管理的简单高效。并且因为是 Gnome 的默认程序，他的全局热键也让很多人趋之若鹜。

离家千里，难免思乡。手里的东西用的熟络了，也不免难以割舍，软件大约也是如此吧。有的人在查皮那里认识得是一个叫做芊芊的音频软件，简洁的界面和强大的功能是她吸引人的地方。好在我们这自由的世界里，也有这么一位类似的佳人，虽然论功能差了一些，但也足够认识芊芊的人追忆往思，她就是 audacious。认识芊芊的人看到 audacious 便可知何为伊人倩影了。

若说功能和外观，音频软件里，当属 Amarok 了。Rhythmbox 的那些功能 Amarok

自然都有，并且，Amarok 英姿飒爽，不免更加惹人喜爱，于是有国人为他开发了插件，其中，便有下载歌词和封面的，这会自然是中文的了。只不过他身为一个 K 派，在 Gnome 环境下似乎总是难免有些碍手碍脚，不得施展。不过别急，G 帮同样不乏精英，那就是有号称 G 帮的 Amarok 之称的 Exaile。Exaile 追求着 Amarok 的追求，感悟着 Amarok 的感悟，同样的精美，同样的能力，让他们虽不在同一个门派，却有着同一个梦想。

说到梦，SongBird 似乎是播放器里面最梦幻的了。梦幻的，让人都不知道是不是该把他叫做播放器。他很苹果，他很 iTune，他很不像一个音乐播放器，但是他却真真切切的如一只青翠的小鸟般在吟唱着清脆的音流。他还集成一个浏览器，让你半随着袅袅只音徜徉在浩浩网际。当看到网页上有优秀的音乐时，他还可以帮你将它收入囊中，伴你左右。

除此之外，还有 Banshee，还有 Juk，还有 LMP，Minirok，Xnosie……太多太多了，一时介绍不完。不知道今天是不是受主人和 MM 的气氛传染，我的程序似乎跑得有些缠绵，哎，大约该把自己重启一下了吧。

故事外的事——解码器

这节咱说了视频文件有编码，播放器要用解码器。这个视频文件都是有一定的编码方式的。比如大家都听说过 MPEG 吧，就是 Moving Picture Experts Group ，动态图像专家组 ，听这名字本来是用来指代一小撮明白真相的群众的，不过后来这一小撮群众发布的标准被广泛使用，于是 MPEG 就成了指代这一小撮群众定义出的那一大撮标准的名词了。MPEG-1是小撮群众在1992年定义出的一个标准，是一种视频和音频的编码方式。大家记得以前的 VCD 不，VCD 光盘上的视频和音频用的就是 MPEG-1这种编码标准。而 MPEG-1标准中关于音频的部分——MPEG-1 Layer3 更是成为了互联网上以及大家口袋里最常见的音频标准——mp3。后来，1994年，这一下小撮明白真相的群众又发布了 MPEG-2标准。MPEG-2向下兼容 MPEG-1，并增加对隔行扫描的支持，被应用于有线电视，还有 DVD 的音频视频编码。再后来，这一次小撮群众又开发了 MPEG3，注意 MPEG3跟我们的 mp3没有任何关系，而且，MPEG3最终没有很好的应用，因为当时人们发现 MPEG2足够了，MPEG3并没有提供足够好的改进。而1998发布的 MPEG4就不一样了，它可以让视频文件的体积更小，压缩率更高，因此得到了广泛的使用。现在市场上卖的 mp4播放器，就是用来播放 MPEG4压缩的视频文件的设备。所以，MP4跟 MPEG4有关，而 MP3跟 MPEG3无关。

说了这么多，回过头来说说解码。视频文件都进行了一定的编码，比如 mpeg-2，或者 mpeg-4。就是说这个视频文件里面的东西都是一大堆乱七八糟的数字，要想看这个视频文件，就得解码，也就是根据这一大堆数字算出应该显示的一帧一帧的图像，并且把这些图像连续播放起来，从而还原成视频。那么这个解码的过程就要靠 Mplayer 老先生了。老先生有很多的解码器，也就是有很多的说明手册，上面写了每种编码格式的文件应该怎么计算，怎么解码。那么以前没有硬件解码的时候，Mplayer 老先生是怎么做

的呢？首先，拿到一个视频文件，然后看看是什么编码的，对着自己的手册，开始解码。解码的过程就是计算的过程，计算需要用什么？好那位同学回答了，得用 CPU 啊。于是 Mplayer 一手拿着手册，一手拎着数据找到我，请求使用 CPU。我说，好的，你就排在 GIMP 的后面，等他用完了你用。过一会 GIMP 用完了 CPU，Mplayer 过去开始拿 CPU 按着手册上写的算法算他那堆数据。最后算出来，得到了几张图片，就转身把图片给图形部门，让他们去显示。然后再从那个视频文件里拿一些数据，再来排队等着用 CPU。由于视频文件的计算量都很大，尤其是高清视频，尤其的大，所以为了保证主人看的电影不变成带旁白的幻灯片，我就要尽可能多的让 Mplayer 多用 CPU，来保证它他能顺利的加码。于是，每次 Mplayer 一播高清视频，CPU 就总被他占着，搞得别的程序都抱怨。现在他终于学会硬解码了，情况就好多了。当然，光他学会硬解码也不行，关键显卡也得支持，而且驱动还得装好才行，不过这些咱以后再说，先说 Mplayer。会了硬解码之后怎么样呢？再播放视频的时候就是一手拿着手册，一手拎着数据找到我，跟我说要用用显卡。可不是 CPU 了啊，改用显卡了。于是我就很乐意的让它去用了，反正别人也用不着，让它自个玩去吧。于是他就去用显卡算去了。用显卡算和用 CPU 算还不一样，CPU 虽然强大，虽然啥都能算，但是要自己手动算。就是说自己要知道算法（对于 mplayer 来水，算法都在解码器上写着呢。），比如要要算出一帧的视频来，要先用第一个数加上第二个数，再用结果乘以第三个数……之类的。这个加啊，乘啊，都是用 CPU 算的，但是中间的过程是要软件（也就是 Mplayer）自己控制的。可是用显卡解码就不一样了，人家那东西是专门解视频的啊，所以你只要把数据放里面，直接就能给你算出一帧帧的画面来。全自动啊！于是 Mplayer 不但不用跟别的软件抢 CPU 了，而且解码的速度还快了不少。

3.4 我的生活色彩

今天一起床就接到了一个任务，挺起还还听轻松，一般胡同里大妈大婶的，经常做这项工作，并且乐此不疲，这就是——串门。不过我去串门可不是聊天取得，我是去做搬运工。在我来之前，主人有许多照片存在了查皮那屋里，而现在可能他觉得这么珍贵的回忆全都堆在查皮那杂乱无章（在我看来）的屋里很不保险，所以，要我去复制一份，放在我这里。于是，我终于有机会去查皮那屋里看看他和他的同志们都长啥样了。

推门进入查皮的房间，里面所有的软件都在睡觉。我看了一下，这个房间是 NTFS 格式的，基本上，查皮只会两种文件系统——FAT32和 NTFS。FAT32是一种很老旧的格式了，连4G 以上的文件都不支持，性能也不好，还不支持多用户的权限，所以基本不怎么用了，多数查皮都会用 NTFS 格式。要说以前，我们 linux 是不太能读懂 NTFS 格式的磁盘的，毕竟是微软私有的格式，我的前辈们基本上只能勉强自从 NTFS 的磁盘上读取东西，往里写是不行的。不过自从 Canonical 学校为我们增加了一本 ntfs-3g 教材以后，读写 NTFS 就都不在话下了。不过虽然能够读懂，但是我自己是不会用这个文件系统的，我会用很多其他的文件格式，比如 ext2,ext3,xfs,jfs,reiserfs,ufs,zfs 等等，各有优势，我现在的屋里使用的是非常强大的 xfs 格式，至于怎么强大，以后慢慢细聊，现在，我要干活了。

来到查皮的屋里后才发现，查皮的屋子里是满地的碎片阿。别误会，这碎片并不是查皮跟老婆打架摔的，而是磁盘碎片，可不是陶瓷盘子的碎片阿。是这样，查皮使用的文件系统是比较容易产生磁盘碎片，隔三差五的就需要整理，可能是我的主人比较懒吧，所以这屋里有很多碎片没有整理。不管这些了，赶紧干活，搬东西。按照主人的指示，我找到了那些照片文件的目录，里面还有很多的子目录，叫什么 bizhi,美女,风景……等等，还有个目录，叫做"..."，这里面是啥呢？打开里面一看，还有一个目录，叫做 XX 门……

总算是把所有照片搬回来了，还没喘口气呢，我们屋里那扇 USB 门上的灯又亮了。

这扇 USB 门是我们与外界交流的一个接口，每次门上的红灯亮起，就说明有东西接到 USB 上了，我就得去打开门看看。有时候门外是一个小集装箱似的屋子，很小，

一般只有几百兆到几个 G 大小，里面也像我屋子里一样放着一些文件。这个时候一般主人就是要让我搬东西，不是把小屋子里的往大屋里搬，就是从大屋往小屋挪。有时候一开门，外面不是一个屋子，而是一台设备，比如是一个鼠标啊，或者是个摄像头什么的，那就是让我去操作这些设备了。主人就接进来过摄像头，摄像头的名字叫 Moto E6，听说这其实是个手机，上面的摄像头可以通过 USB 接口来给电脑用，不管怎么样吧，这东西咱也会玩，接上就能用，鼠标那就更不在话下了。不过总的来说，还是往 USB 那门外接小集装箱屋子的情况多，这种小集装箱式的屋子叫做 U 盘，也有更大一点的，就是移动硬盘了。每当 U 盘接进来的时候，我就把它当作我屋子的一部分来用。

这回主人接到 USB 上的又是一个集装箱式的空间，不过还比较大，4个 G。我仔细看了看，结果没有发现这设备的名字，于是我就直接给他起个名字，并在门上挂上了牌子：/media/4.1G ，就像我屋里的/home 分区一样，我屋里的所用分区，都必须有个牌子。这个挂牌子的过程，用我们的专业话说，叫做挂载，这个大家装系统的时候应该都听说过哈。挂载，就是挂牌，就是在某一间屋子的门口挂个牌子，起个名字，到时候干活的时候就好说话了。直接说屋子的名字就好了。名字，或者说牌子，就是个标志，是可以随便换的。比如有个分区被挂载到了/home/目录，也就是说，有个屋子 A（就叫 A 吧），被挂上了/home/的牌子。那么主人说要看看/home/下都有什么，我就会把那个屋子 A 里面的东西列出一个单自来给主人看。等回头可能又把别的分区挂载到/home/下了，那很简单，就是把/home/那个牌子从 A 房间门口摘下来挂在 B 房间门口而已。主人再说要看卡/home/下都有什么，我就该把屋子 B 里面的东西列表来给主人看了。没啥经验的主人可能会大跌眼镜：哇塞～怎么我/home/下原来那些东西都没了阿！！都哪去了阿！！ 殊不知，其实原来那些东西还在 A 屋子里好好的放着，只是现在 B 房间改叫/home/了而已。

又废了这么些个话，不好意思。回来说这回接上的这个4G 的屋子，接上之后，还没等主人发话，我先去里面查看了一下，一看全都是些个 jpg 的文件，我是看不懂这些个文件，但是我知道有人能懂，于是赶快通过图形界面那哥几个报告主人：您插进来的这个里面貌似全是照片，是不是要我给您找来 F-Spot 呢？。

F-Spot 是我们这里的一个管理照片的程序，他可以帮主人把移动设备（就是那种集装箱似的小空间）里的照片导到硬盘里，并且按时间分门别类的管理好。主人想要去年三月的照片，他马上能够给找到；要前年5.1假期的，没问题；要看宋朝的……给把铲子自己挖去吧。除此之外，还能够汇报照片的信息，比如用什么相机照的，照的是后用的光圈，快门，ISO 都是多少，等等信息。这道不是 F-Spot 厉害，能够光看照片就知道用什么相机照的，而是因为相机在照这张照片的时候就把各种参数写进照片文件里了。

主人似乎对 F-spot 的能力还不大熟悉，于是没有让我去叫醒 F-spot 来导入照片，而是自己手动把 U 盘里面的照片拷贝到了那个放照片的目录。考完之后，主人看了看图片目录的文件夹："bizhi", "美女", "风景"……还有从 U 盘拷过来的"静物",恩……都齐了。接着，就下下达了卸载 U 盘的命令。有人可能问了，你主人怎么没看见那个"..."目录以及里面的 XX 门照片？这个要解释一下了，在查皮那里，文件和文件夹的属性中有一项是隐藏，隐藏的文件一般是看不到的。而在我们 Linux 这里，文件是没有隐藏这个属性的，但是也有办法让一个文件或文件夹在默认情况下不显示出来，那就是给这个文件起一个以"."开头的名字，那样的话，无论自终端里 ls，还是用"鹦鹉螺"来查看，都看不到这个文件了。要想看到也容易，在终端里就 ls -a，在图形界面的鹦鹉螺里就按 ctrl-h。

不知道主人是有意的还是无心的，反正那个目录其名叫做...，他自然就看不到了。

照片拷进来后，主人让狐狸妹妹去找找好用的照片管理软件，终于从狗狗那个无敌的搜索引擎里知道了，原来我们 ubuntu 本来就带有 F-Spot 来管理照片。绕了一圈终于绕回来了，主人下令：运行 F-Spot。哎，你说你刚才拷照片的时候就用他多好。F-Spot 被我叫醒之后，赶快进入工作状态，由于是第一次运行，他上来就报告：现在还没有导入任何照片，请主人选择要导入的照片。这话说的在理阿，他就是个管理照片的，你得赶紧跟他说让她管理那些照片阿，要不然启动了他也没事干。可是主人似乎对这个"导入"有些迷茫，心说了，我把照片都放在“图片”那个目录下了阿，你不会自己找找么？得，你让我导入，我就导入把，于是主人在 F－Spot 给出的导入窗口中选择了"选择文件夹"，同时他还看到了"（未检测到相机）"这么个不可选的项，看来如果有相机接进来的话是可以直接从相机中导入照片的。主人选择了图片目录下的"bizhi","美女","风景"等几个文件夹来导入，F-Spot 立刻工作起来，不一会，主人就在 F-Spot 的主界面上看到了自己导入的那些图片，并且已经按照时间顺序整理好了。选中一张图片，还能在左侧显示详细信息。

研究了一会，主人觉得这软件还算比较好用。关了之后，打开了图片文件夹，一打开愣了——怎么在图片文件夹里面多了个"照片"文件夹呢？赶紧进入照片文件夹看看，发现里面是以拍摄时间创建了很多级的目录，最终的目录里就是原来那些 bizhi 阿,美女阿的文件夹里的照片。原来，F-Spot 的正常使用流程是从相机或者存储设备中"导入"照片，这导入的过程其实说白了就是把照片从这些设备中拷贝到本地硬盘上，F-Spot 会在图片目录下建立照片目录，所有导入的照片就都在这底下，可是主人不知道阿，结果就把本地磁盘上的图片导入了一边，于是每个图片就都有了两份，图片目录下一份，照片目录下一份。这一下然主人觉得很不爽：这是什么破软家阿，原来是这么个导入法阿，太弱智啦～！

暴走一阵之后，主人重新坐下来叫过狐狸妹妹，再找找管理照片的软件。其实我们 Linux 下用来管理照片的软件确实少一些，但是不乏精品。首先说 F-spot 能力就还是不错的，只是主人不大习惯他的方式，呵呵。另外还有 digikam，是 K 派那边的强大的照片管理软件，比 F-Spot 还要强大，不过这话可不能当着 F-Spot 说，要不又吵起来了。主人找了一会，终于作出了自己的选择，叫来超级牛力，安装软件。只听得超级牛力高喊着：“本 APT 有超级牛力～～～”就跑出去了。我问 ibus：“刚才主人给超级牛力输入了什么？”

“报告头儿，是 Picasa”

“星爷，查查这啥意思。” 星爷就是星际译王，一个字典。

“这个吗……英法美德俄日意奥的语系里都没这个词。不过有一个长得比较像的。”

“什么？”

“Picasso ，毕加索。”

“哦……看来主人是要换个管理照片的软件了。”

F-Spot 不服道：“管理照片？难道我刚才的表现不好么？？”

我只得双手平摊做无奈状……

数分钟后，超级牛力归来，带来了一个穿的花花绿绿,很有艺术气息的家伙,等等，这哥们身上的颜色怎么好像在哪见过？我过去上看下看左看右看，怎么就看着不像我们这的人呢？于是我叫来了 file。 file 可不是一个普通的文件，而是一个程序，一个用于判断文件类型的程序。他可以判断一个文件是什么类型的文件，当然也能判断可执行的

程序。他可不是跟据扩展名来判断，叫.jpg 的就是 jpg 文件，叫.txt 的就是文档文件，这种功能，连傻子都会。file 的功能要强大的多，他是根据文件的内容来判断的。一般一个文件都会有个文件头，来说明这个文件的 类型。比如 JPEG 类型的图片文件，他的文件开头的两个字节肯定是 FFD8（16进制），而 GIF 文件的文件头就是4749463839，其实就是 GIF89几个字的 ASCII 码。二进制程序也有类似的特征码，于是，我让 file 赶快去看看这个“毕加索”（就叫他毕加索吧，虽然还是差了几个字母）到底是个什么程序。file 把毕加索上上下下的检查了一遍，得出结论——这是个 Windows 的 EXE 格式的程序。

“什么？Windows 的程序！？超级牛力啊，你别是走错了吧，怎么把 windows 的程序领来了？”超级牛力不急不慌的摇摇头：“本 APT 有超级牛力， 怎么会搞错呢，这个就是从源里找来的软件包。不过别急，本 APT 有超级牛力，这软件包可不是光毕加索一个，后面还有一个呢。”我这时才注意到，老毕后面还站着一个家伙，这人又是谁呢？没事，超级牛力哪里肯定有这个软件的资料，让他查查吧。还没等我让他查呢，他已经向大家解释上了：“毕加索先生是 Windows 界成名的图片管理大师，他所在的公司，也就是狗狗哥那公司，他们公司为了惠及 Linux 世界的人们，又为了偷懒，把毕大师配上一个翻译就直接推向了 Linux 界。”哦，原来这样，我说怎么看着这身花里胡哨的颜色眼熟呢，因为他跟那个 Chrome 的颜色差不多，都是狗狗公司的嘛。那么后面那个既然是专门负责给毕大师当翻译的，我们就叫他毕翻译吧。

毕大师和毕翻译安顿好之后，主人没有立刻把他们叫起来干活，他似乎有些什么别的事情，装完了 Picasa 就离开了。但是毕大师及其翻译并没有消停，俩人先后被 F－Spot 拽进内存，f-spot 是一直跟着我的软件，我对他的脾气很了解，有点太直，他这种元老级的软件被主人换下来自然不爽。一直以来都是他负责管理照片的，论能力他自觉得不输给任何人。现在主人刚用了他一次就找来这么个功能差不多的家伙来代替他， 这不是明摆着砸他 F-Spot 的牌子么。要是以后让这个 windows 的程序代替了，我们 linux 程序的脸面还往哪搁？于是 f-spot 决定，为了荣誉，向毕加索挑战！只见 f-spot 跑到被拉进工作间的毕大师面前说：“大师既然出身名门，本领自该不弱。在下不才，愿在大师面前领教几招粗浅功夫，不知道大师可肯赐教否？”只见毕大师的表情如平静的湖水般并没有因 f-spot 的挑战而激起一丝波澜，只是面容祥和的扭过头对翻译说：“What did he say?” 靠……

经过毕翻译的解释，毕大师明白了 F-Spot 的意思，微微一笑，做了个请的手势。我一看，得，又斗上了。

大激突！

要比就从起床开始比！f-spot 和毕加索以及毕翻译重新回到硬盘睡觉，然后我去叫来的 time 同志。time 是一个用于计时的命令，这个咱以后再说，先 看比赛。随着我的

一声号令下，time 开始计时。f-spot 蹦起来后牙也不刷，脸也不洗（废话，一个软件，有牙么？），迅速的从硬盘飞奔进内存。再看那 边，毕翻译先迅速跑进了内存，然后再扭头去叫醒毕大师——因为毕大师听不懂我们的话，所以无论我们怎么喊都是叫不醒他的，只能先叫醒翻译，再由翻译去叫醒他。这样一来，时间自然慢了不少，对于起床速度，F-Spot 完美胜出。双方起床已毕，相向而立，只见 F-spot 掏出两张一模一样的照片，照片上是一个 人像，似乎是晚上照的，眼睛如含着血泪般发出令人不寒而栗的红色。只见 F-spot 把一张照片扔给毕大师，另一张贴在自己这边，双掌运足力气，瞄准照片中人的双眼大喊一声：嗨！立时，照片上人的红眼不见，翻了白眼。另一边的毕大师微微一笑，拿起自己这边这张，单掌向前一推，一股掌风直逼那人双眼，只见掌风过后，那人双眼渐渐恢复成正常颜色。F-spot 不等毕大师打完那掌，又拿起照片推拳运功，只见那本是夜里的照片亮如白昼。毕大师也不示弱，将照片抛向空 中，双手一抖，一道劲风吹过，再看落下来的照片时，也已经比原来明亮不少。F-Spot 又对照片连续发力，打出三招，依次改变了照片的对比度，色调和饱和度。毕大师口念咒语：“That's easy……”只出一招，双手间出一道白气，就把照片的亮度，对比度，色调，饱和度，都改到合适的状态。毕翻译的在旁边解释道：“ 这招乃是毕大师的独门秘诀，叫做‘手气不错’！”毕大师微微点头，一扬手，只见那修改好的照片激射而出，直接从网口飞了出去，发布到了 PicasaWeb 网站上。屋内众人顿时为 F-Spot 捏一把汗，大师叫 Picasa,那这 PicasaWeb 网站，明显是人家地盘啊，F-Spot 能搞定么？哪知道 F-Spot 不慌不忙，也照片 往网口一扔，把照片同时发布到了 Flickr,PicasaWeb 等多个网上相片储存空间里。这真是棋逢对手阿，正闹腾着呢，主人回来了，听声音后面似乎还有一个人，等她一说话我听出来了，就是上次那个 mm。

主人似乎是和 mm 欣赏照片来的，只见主人打开了 Picasa，毕大师和毕翻译跟 F-Spot 斗完了还没来得及回硬盘呢，正好就直接投入工作了。主人还夸 Picasa 启动很快，恨的 F-Spot 牙根痒痒——他启动能不快么，压根就在内存里呢。毕大师第一次启动，要扫描主人的整个目录，主人也没在意就直接允许了，然后毕大师就开始一个目录一个目录的搜索，先是"bizhi",然后是"美女"，之后的"风景"……"XX 门"——等等！当着 mm 看这个不大好吧……

当初把这个目录拷贝过来的时候还以为是主人为了隐藏，故意起名以.开头呢，和着他自己也没注意到这个目录阿。也是，拷贝过来之后就隐藏了，他当然看不见，F-Spot 是需要手动导入照片的，导入哪个目录才能看到哪些图片，不导入的看不到。可毕大师可不一样，他可是搜索你目录下所有的图片，不管什么隐藏不隐藏。话说回来了，毕大师本来就是 windows 的软件，他们可不知道“.”开头的软件是隐藏的阿。结果毕大师很实在的把那个...目录下的 XX 门目录的图片也都给扫描了。毕大师一边扫描，会一边再屏幕的右下部分显示出当前扫描到的图片的缩略图。虽然每张图片都是一闪而过，主人似乎还是看到了很么，赶快把毕大师最小化了,可是毕大师似乎并没有理解主人的意图，主窗口虽然最小化了，但是依然在屏幕的右下方显示着正在扫描的图片，不过索性毕大师扫描的很快，一转眼的功夫已经开始扫描下一个目录了，主人的心才算放下去一点，正这时候听那 MM 说：照片呢？快让我看看吧。

主人似乎是迟疑了一会，到底是该打开毕大师来浏览图片还是去那个目录下一张一张手动打开看。经过了复杂的思想斗争后，终于决定——把毕大师关了！然后打开鹦鹉螺，找到存放这照片的那个文件夹打开来看，安全第一！

来到存放着主人照片的文件夹，只听 mm 惊呼：“咦？照片都是这样显示的阿，真

先进。”主人得意的点点头：“那是那是。”心里可一直打鼓：“这么手动一张一张看，会不会被鄙视连个 ACDSee 那样的软件都没有呢”但反正也没有退路了，主人也就只好双击了一张照片。按照默认的设置，我去叫醒了 Eye of Gnome,中文叫做 GNOME 之眼。这家伙是一个用来浏览图片的软件，功能上相当于查皮那里自带的图片查看器。Gnome 之眼（咱就简称他 G 大眼吧）支持基本所上所有常见的图片格式，可以对图片进行缩放和旋转。主人看到了 G 大眼的界面，虽然简洁，倒也算像那么回事，这才放心了。他一边和 MM 评论着照片，一边按向右的按键查看下一张。这个目录下的照片是从 U 盘拷过来的那堆照片之中的一部分，似乎是主人和 mm 出去玩的时候照的，俩人一边看，一边评论。看到一张照片有些问题，里面的人都红眼了，这是晚上照相的时候经常遇到的。主人顺手点击 G 大眼界面右上角的"编辑图像"按钮，G 大眼就叫来 F-Spot 来对图像进行编辑。这一下 F-Spot 和算是有了一种看见伯乐的感觉了。G 大眼和他都是 Gnome 自带的标准软件，是 G 帮的招牌，俩人配合起来那真是天衣无缝。F-Spot 在左侧列出了可以对这张照片进行的处理，主人选择了“降低红眼”，很方便的就修复了过来。继续往下看，有的照片是相机竖着照的，需要旋转一下，这事就不必惊动 F-Spot，G 大眼自己就给解决了。又看了几张，主人觉得一张一张往下翻太麻烦，想起在查皮底下用 ACDSee 的时候可以按 F5自动播放，于是抱着碰碰运气的心里也在 G 大眼的界面上按了一下 F5，果然 G 大眼就进去全屏播放的状态。这种状态下，手动翻页也可以，不翻页的话，到一定时间自动显示下一张。

说说笑笑中，时间很快就过去了，现在时间是19：00分00秒，距离晚饭还有1412秒。今天主人与 MM 进行了亲切的会谈，会谈中，宾主双方就哪些照片该删，那些照片照虚了等原则性问题，达成了高度一致的共识。MM 高度赞赏了主人操作电脑的能力，主人也表示将继续坚持两个系统一起抓，两手都要硬的原则，并对 MM 能够一贯相信世界只有一个系统，微软是计算机不可分割的一部分的想法表示万分哀悼。会谈中，主人还指出，Ubuntu 是 Linux 中的一员，在当今世界形式下，应团结所有 Linux 世界发行版力量，为把自由软件建设成为富强，繁荣，发展的社会主义新软件而努力奋斗。会谈结束后，主人邀请 MM 一行共进晚餐，餐前没忘了把我关了……

晚上，主人独自晚餐归来，一回家就赶紧把我叫醒，启动之后第一个命令就是去找来毕大师，我明白主人是想看看那个 XX 门的图片到底是怎么回事，为什么之前一直没有注意到，却被毕大师搜索出来了。毕大师起床后在主界面按时间顺序列出了自己搜索到的图片，主人很快的找到了"..."目录下的 XX 门目录，看见里面那一百多张图片，不禁血压上升，只见每张图片拍摄角度各不相同，色彩灯光各有差异，但上面都无一例外的有两个主要元素——一扇孤独的铁门，和一行“插插防盗门，插上不进人”的广告语……主人泪奔中。

自从 F－Spot 和 G 大眼的配合获得主人的认可之后，F－Spot 的心里总算是平衡了不少，不过平时主人整理照片还是主要用毕大师，F-Spot 只是用来顺手处理一下照片。可是这 F-Spot 毕竟是专业管理照片的，不是专业处理照片的，就算他能通过安装插件来扩展对照片处理的效果，可毕竟能力还是有限。于是主人通过明察暗访，终于知道了一个响当当的名字，一个更加前大的图片处理软件——GIMP。

GIMP 这家伙是个天才的美术家，是我们 Linux 界有名的图像处理软件，号称 Linux 的 PS。他能够画出很多漂亮的图画来。当然，漂亮不漂亮是以主人的审美观点来说的，对我来说他制作出来的那些个东西，不过是一个写满0101001的文件而已，和其他的文件没有什么区别。虽然他可以在一张白纸上创造出一个多彩的世界，但多数情况下他要

做的工作是调整一个已经画好的图片。比如调整一下图片的亮度啊，色彩啊什么的。有时候还让 GIMP 做一些效果，比如做成油画效果的，就是把照片做成像是画的油画似的；还有做成浮雕样的，或者加个相框，他都行。更高级一些的，比如在树林里 PS 出一只华南虎阿，把图片里路过的路人甲换成小胖之类的，也都没有问题。同时 GIMP 也像狐狸妹妹一样，可以安装很多的插件以实现各种不同的功能和效果，论功能，不输于查皮那里的 PhotoShop。但是 GIMP 有些不大好接触，总是按着自己的性子来，让主人用起来有些不顺手，不如 PS 那么易用。毕竟人家 PS 身价不菲嘛，听说一套要好几千块钱，一分钱一份货嘛。花这么多钱请回家去的软件，怎能不服服帖帖的听主人的话呢。再看看 GIMP，总是摆着一副：我就这样，爱用不用的架势。也许就是因为他太过冷酷，原本默认安装在系统里的 Gimp 从我们这一届 Ubuntu 开始，不再默认安装了，毕竟多数用户用不到这么专业的软件。如果想用他也不难，找超级牛力装上就好了。

主人得知有这么个软件后，赶忙装上体验一下。他首先用 Gimp 打开了一张照片，试着调整一下亮度，对比度，拉了拉曲线，觉得不如 PhotoShop 好用，论易用性自然是差些，论起功能来，也就只相当于30%的 PhotoShop，不过也算是够用了，以后就可以完全在 Ubuntu 里处理照片了。

3.5 我的办公软件

今天，主人打开电脑，破天荒的先去叫醒了 OO 老先生，往常都是先去叫醒狐狸妹妹或者去找那个 World of Goo 玩一会。今天找 OO 老先生干什么呢？

OO 老先生知道是谁吧？就是 OpenOffice.org，之前给 SMplayer 和 Gnome Player 劝架的那个。我们喜欢叫他 OO,因为他待着一副很有学问的眼镜，而他的全名，叫做 OpenOffice.org——相信我，这确实是个软件的名字，当然，同时还是个网站的名字。OO 老先生是我们这里的老前辈，他是我们这里的文书，主人要写点什么啦，做个什么报表啦，演示文档啦之类的，都找他。今天主人找到他，打开一个空白文档，写下两个大字——简历。

原来主人是要找工作了阿，这可是个麻烦的事情，他们人类找工作可麻烦了，又得简历又得面试什么的，还一面二面三面四面，五六七八面。哪像我们操作系统，看着顺眼就往硬盘上一装，简单。主人写下了头两个字后，便呆呆的等了半天，也不知道干什么呢，等了大约5分钟，终于有下一个动作了——叫醒狐狸妹妹。然后主人牵着狐狸就去网上找简历模板了，看来还是个新手，呵呵。要说模板，OO 老先生就有很多的模板，文字的也好，幻灯片的也好，表格的也好。但是默认并没有安装这么多，你可以找到在背后支持 OO 老先生的社区去找，那里有很多的模板资源，地址就是这 http://opentemplate.org/ 不过主人似乎并不了解这些，还是依靠狐狸妹妹去找到了一写简历模板，照着写起来。

主人这写着，OO 老先生一边看着一边评论："恩……这句不合适""这个字体不好看，怎么用这个呢？"，"恩，这段不错。"OO 老先生在这里评论着，把另外一位有学问的招来了，他叫做星际译王，之前也出现过，我们都叫他星爷（可不是周星驰阿）。

星爷来自中国，是我们这里少数来自中国的软件之一。他可以说是我们这里最有学问的人，简直是什么都知道。一开始他只是懂点英语而已，大家都只当他是个英语翻译，后来主人叫狐狸妹妹去给他找来各种各样的书给星爷看——就是那种叫做字典的书，这种书只有星爷能看懂。看了这些书以后，星爷知道的就多了，什么日语啊，法语啊，德语啊，都会了，估计联合国要开会请他一个人去当翻译就行了。于是星爷从英语翻译一下子晋升成为了地球语翻译！（地球上的语言都会-_-b）不知道他以后会不会再学学火星语？

然而星爷是不仅仅满足于当地球语翻译的，或者说，主人是不满足于让星爷只做个翻译的。这之后他又给星爷找来了本《陈义孝佛学常见辞汇》，于是星爷开始研究佛学

了，不过这东西只是主人一时的好奇而已，后来就再没给星爷找过佛学的书，而是找了本《五笔98》，开始学习五笔了。要说这输入法的事，那可是 ibus 的本行啊，可是无奈 ibus 老弟干活还行，表达能力不强。主人要打什么字，ibus 可能很快反应过来，打在屏幕上，可是主人要是忘了某个字怎么打，问问 ibus，那可要了命了，打死他也说不清楚啊。主人只能问："是 a 开头不是？"然后 ibus 把所以 a 开头的显示出来数一遍，摇摇头："不是"，然后主人再猜："是 s 打头？"ibus 再把 s 打头的列一遍……这样实在太费事，于是主人就让星爷学五笔，到时候就能去问星爷："这个……'我'字用五笔怎么打啊？"星爷会投过一个鄙视的眼神："你还好意思说会五笔啊，q 空格！"

会了五笔还不算完，后来星爷又看起了《本草纲目》，研究起了中医。不过他还不能当大夫，针灸号脉啥的他是不会（就算会也没法号啊），那会啥呢？看了《本草》当然最精通的就是药理了，随便说一种药，他就能告诉你这个药的药性，药效，怎么用，等等等等。这时候基本上我们已经对星爷的全能感到震惊了。后来他又开始研究古汉语，看古汉语词典，康熙字典，整天到晚的跟我们这之乎者也。"夫内核者，老大也，发其命，出其令，而统'康皮右特儿'（computer -_-b）……"然后就是我们集体的"打豆豆"时间——谁是豆豆就不用说了吧。

我也趁没人的时候，问过星爷：您怎么懂这么多呢？看什么会什么。星爷很神秘的笑笑说：其实他没什么本事，就是在学校学过信息检索而已。主人给了那么多本书，要想都记住，根本不可能嘛，他只是每次在主人问起事情的时候，赶快现去查书，用最快的速度找到并告诉主人。要是没了这些书，他知道的就少很多了。不过也不至于离开了书就什么都不知道，现在的星爷学会上网了，可以连接到一种叫网络辞典的地方，自己不会可以去网上查，不过那样效率自然不如自己翻书快了。

这时候主人终于凑合出了一篇中文的简历，马上还要再写一篇英文的，这一下 OO 老先生自然还要继续工作，星爷也被调动起来——查词阿，那是一个词一查阿，星爷十分怀疑主人的四级是怎么过的，心说你给我装这么多本字典自己怎么不先好好看看。可是抱怨也没这，软件的天职就是运行嘛。连查带编，总算是把英文的简历也凑出来了，尽管英国人不一定看得明白，但是有一个总比没有好不是。写完了之后赶紧让 OO 老先生存起来，老先生其实在主人写的过程中就不断的在往硬盘存了，防止忽然掉电嘛，这都是跟查皮里面的那个 Office 学的。现在主人要存，OO 老问明白了存的地址和文件名，就把文件存好了，然而存好了之后他微微笑笑，自言自语地说：这个早晚还得重写……

简历不是存起来看着玩的，得发出去给别人看才行，这不主人正在用"心有灵犀"登录他的 msn 帐号，然后给别人传简历——呃……这个方法是不是不大正规阿，哪个公司这么收简历的。我问心有灵犀："主人在给哪个公司发简历阿？"心有灵犀说："咳，哪有公司这么收简历的，主人是在给前几天来的那个 mm 发呢。"哦……怪不得呢，原来主人只是在交流写简历的经验阿。我说："估计是 MM 不会写，要拿主人的来看看。"心有灵犀却并不关注主人的目的："我虽然不知道为什么主人给她发，但是我知道，那 MM 肯定看不了主人写的这个简历。"我正要问为什么，这时候，有一个人在旁边冷不丁的说了一句："哼，干吗不用邮件传呢。"我一看，是 Evolution，我们这里的信使，默认的电子邮件软件。

说起电子邮件，大概您不会陌生。不过要说邮件客户端，可能在公司上班的还会经常用，在家里用的就很少了。一般上网的人虽然都会有自己的邮箱，但大都是通过网页来首发邮件，用邮件客户端的人比较少。而我们 Ubuntu 又很少会用户工作环境（尤其是国内），于是，Evolution 也就成了我们这里最怀才不遇的人。其实 Evolution 的本领

还是可以的，他的工作基本上就相当于查皮那里的 OutLook，除了收发邮件，还可以安排日程。阅读邮件的时候，Evolution 也基本生什么样子的 邮件都可以正常显示，中文英文的也都没问题。不过还是有很多狐狸妹妹的粉丝没不使用 Evolution，而是安装了狐狸的同门师姐——雷鸟姐姐

雷鸟姐姐（ThunderBird）和狐狸妹妹有很多共同之处。比如师出同门，都是 Mozilla 的得意门人；比如都可以安装丰富多彩的插件；比如都因为在 Windows 平台下良好的表现而拥有很高的知名度。所以很多人慕名而来使用了雷鸟作为邮件客户端。雷鸟姐姐能力确实也不再 Evolution 之下，邮件的收发，显示，自不必说，关键她长的也漂亮，而且简洁高效，符合 Linux 的哲学。可 Evolution 也有他独到的地方，这就是和 Exchange 的配合。很多公司的邮件服务器都使用了有点软公司的 Exchange Server 作为邮件系统。除了他们自家的 OutLook 外，像雷鸟姐姐这些开源的客户端多数都不能够正常的连接到 Exchange 上，然而 Evolution 除外。经过多年修炼，Evolution 顿悟了 Exchange 的链接原理，他默认情况下就可以正常的与 Exchange 进行链接，首发邮件，这也是为什么他成为了 Gnome 里面默认的邮件软件的原因。

说了这么多，可惜主人只是一个普通的家庭用户，并不需要一个邮件客户端，所以 Evolution 在我这里从来就没运行过，也难怪他没事就发牢骚呢。

正这时候，心有灵犀笑着对我说："你看怎么样，我就说这文件发过去也看不了吧"我一看，果然，那 MM 正跟主人抱怨呢："你发的是什么文件阿，打不开。"我明白了，主人用 OO 老先生写完了简历，顺手就保存成了 OO 老先生默认的 odt 格式，这虽然是个开放的格式，可是查皮那里的 Office 只有2010版本的才能打开，这之前的版本都是不能打开 odt 格式的。估计 MM 那里的 Office 就是不支持 odt 的老版本 Office，所以无法打开。要我说这就是查皮那个 Office 装蒜，就是诚心不愿意支持这个 odt 格式。你想，一个开放的格式，又不需要破解，要想支持早就能支持了，何必非得等到快2012了才来支持这格式。不过也没关系，OO 老先生也可以把文件存储为 doc 格式的，那你总不能再打不开了吧。主人很快也意识到了这一点，于是把那个文档重新打开，另存为.doc 格式，再次发给 MM，MM 这回倒是看到了，可是问题又来了："怎么格式有点乱呢？"

唉，要说这是在不能怪 OO 老先生了，这个 doc 格式是那个有点软公司自己设计的，并且还不开放，别人是不知道这个 doc 格式的文件应该怎么写的。可是 OO 老先生经过多年的潜心研究，分析过很多 doc 文件，总算是研究除了这种格式的大致写法。可是毕竟只是研究出了一部分而已，如果文档里使用了一些复杂的结构，OO 老先生可能就搞不定了。于是文档的格式和字体等和查皮的 Office 里看见的有些出入也是可以理解的。可是我理解没用，MM 不理解不行阿，就算 MM 理解了，这简历发到人家公司的人事部去，人家一看这格式乱七八糟的人家也不理解阿。这怎么办呢？说来也没啥好办法，只好用查皮的 Office 去写，或者就把 odt 导出成 pdf 格式再发。OO 老先生要想文件输出成 pdf 还是非常方便的。

那么有没有更能够理解 doc 格式的办公软件呢。还是有的，永中 Office 就是其中之一。永中 office 是一个 java 的程序，像星爷一样来自中国。他的目的就是呈成为能够替代 MS Office 的一体化办公软件。他对 doc 格式文件的研究确实是比 OO 老先生深刻，和独到，基本上可以做到在永中下和在 MS Office 下看到的效果是一致的。（当然偶尔也会有例外）然而可能是由于他是 java 的，运行起来效率不是很高，您听这名字，永中，听着就觉得臃肿。再有，他也不开源，是一个闭源的软件，而且不支持 OO 老先生的 odt 文件，所以家庭 linux 用户使用他的人不是很多。如果你想体验一下的话也很方

便，只要去永中的官方网站去下载，http://www.yozosoft.com/download/zmo.jsp 就好了。下载来是一个压缩包，解压后的目录里面有一个 setup 文件，默认可能没有执行权限，需要手动加上。过程很简单，右键点击该文件，选择属性，在权限标签里勾上“允许以程序执行文件”，就好了。关闭之后双击 setup 文件就开始安装了。

除了 OO 老先生和永中外，我们 Linux 世界里还有很多类似的办公软件。比如 K 派的 KOffice，就是比较有名的一个。虽然很多以 KDE 为主要图形界面的发行版依然会用 OpenOffice 来做默认的办公软件，但其实 KDE 是拥有自己的办公软件 KOffice 的，Koffice 的特点是拥有广泛的工具集，除了文字处理、电子表格、演示文稿和数据库系统之外，KOffice 还包含三个图形处理工具---流程图应用、矢量绘图软件和光栅设计应用 以及项目管理工具、报表开发工具、制图绘图工具和数学公式编辑器。但是 Koffice 对文档格式的支持，尤其是对微软格式的兼容性远不如 Openoffice，更不用说永中了。（Koffice 喊冤：我是 Linux 软件，干吗非得支持微软的格式！）所以 KOffice 使用并不广泛。如果你只是想有一个处理文档的小软件而不是包括了演示文档，电子表格等等功能的办公软件，那有一个小家伙你肯定喜欢——AbiWord。听名字就知道了，他就是个 word 替代软件，而不是什么什么 Office。这小家伙只有十多 M 大小，然而平常所使用的几乎所有的 Word 功能---封、报表、邮件合并、修改跟踪---他都可以在这里找到。AbiWord 并不适合企业级专业人员使用，但是对于那些工作要求不是那么严格的学生和个人来说，AbiWord 足够了。他支持 odt 和 doc 格式，不过兼容性似乎不太好，而且打开包含多种语言的文档时会遇到点问题，但愿随着以后的发展会越来越完善吧。

主人最终还是让 oo 老先生吧简历输出成了 PDF 发送给了 MM，这也算是目前 Linux 和 Windows 平台交换文档时候比较通用的一种方式了。MM 终于看到了主人的格式规范的简历，并且通过心有灵犀，发过来了一句评语：“乱七八糟”。呵呵，看来就像 OO 老先生说的，主人要重写了。

Chapter 4

第四章

虚虚实实

4.1 红酒大师

经过了工作间内激烈的角逐，各类软件的选拔已经尘埃落定。BT 下载类软件中，奔流脱颖而出；视频播放软件里，Smplayer 异军突起；图片管理软件组，picasa 稳操胜券；网络浏览器呢，自然是狐狸妹妹成功卫冕。然而，虽然狐狸胜过了 Chrome 和 Opera 这样的对手，却不知道有更要命的问题在前面等着她。

这一天主人带着狐狸徜徉在网络的海洋中，忽然想起一件事情——好像该发工资了。原来，前一阵子主人写的简历还是起到了些作用，目前正在一家小公司工作。三个月的试用期刚刚过去一个月，主人就惦记起自己的第一笔工资了。主人掏出自己的工资卡，忽然想起来，这存了白存银行在自己家附近好像没有，因为毕竟是小银行嘛。要是坐 8 站汽车走三里路去查查余额呢，又有点不值得，所幸这存了白存银行有网上银行，直接上网查查吧。于是，主人牵着狐狸向前一指——目标，存了白存网上银行，出发！

说时迟，那时快，转眼间狐狸已经跑到了存了白存网上银行。主人勒住司缰向上望去，只见页面上赫然写着四个大字，分上下两行。上面一行写着：帐号，下面一行写着：密码，然而奇怪的是，帐号和密码后面，都没有输入框。主人一时摸不着头脑，仔细一看，底下还有小字：“请安装存了白存网上银行提供的安了白安全插件”。主人顿时醒悟，哦，这银行的网页自然安全系数不一般，既然要装插件，那就装吧。可是当主人按照网页上提供的连接地址把这个插件下载下来后就傻眼了——怎么是个 exe 文件呢。

这个 exe 是 Windows 下的可执行文件，这种软件只能在 Windows 下运行，在我们 Linux 系统下面可是无法工作的。首先就因为语言不通，我们 Linux 软件说的话他们 Windows 软件完全听不懂，反过来也是一样，所以我根本就没法叫这种软件起床，更别说命令他工作了。再说他们 exe 软件运行时需要调用的各种库文件，也就是那些 dll 文件，我们这里也没有，我们这里的库文件都是 .o 格式的，所以 exe 在我们这里是没法直接运行的。这个破银行只提供一个 exe 的插件，这不就是存心不想支持我们 Linux 嘛。主人也迷茫了一下，不知道该怎么办。不过有句话说的好：内事不明问老婆，外事不明问 Google。于是主人赶紧让狐狸去狗狗哥那里问问到底是怎么回事。这一问才知道，原来，这些国内的银行们大都使用了不大安全的 ActiveX 技术来写网银的安全控件。（怪不得叫安了白安呢，安了之后也不安全，自然也就存了白存了）然而这 ActiveX 技术是那个有点软公司发明的，人家是自产自销，只有自己家的 IE 浏览器能够支持这种技术，其他的浏览器想都不用想了。狐狸妹妹的扩展和插件虽然不少，可是没有一个可以让狐狸妹妹支持 ActiveX 的。虽说 Windows 下面的狐狸可以使用 IETab 扩展，可那其实就是调用 IE 来浏览网页了，那得先有 IE 才能用 IETab 扩展，我们 Linux 这要又怎么会有 IE 呢。狐狸妹妹问明白了之后急直跺脚，可着急也没用阿，也只能灰溜溜的回来告诉主人——我实在搞不定这个网站，你就是掐死我，我也没辙。

主人只好陷入了无奈之中，难道为了查看一下工资卡里的余额，竟然还要重启一下去一趟 Windows 么？不行，还得上网搜搜，不可能只有我一个人碰到这个问题吧。于是，主人又牵过来狐狸妹妹去网络上搜索了。主人这边搜着，工作间里面也没闲着，大家七嘴八舌的议论起来……

今儿个丢人丢大了！

想我大名鼎鼎的火狐狸，什么样儿的网站没去过？什么样儿的网页没显示过？什么 Konqueror，Epiphany，lynx，除了那个自以为是的挪威妞儿以外，哪个浏览器见了我不得叫声大姐？你说是在线视频还是 Flash 游戏，是 IPV6 还是数字证书，咱哪点儿含糊过？再说有我一身的插件和扩展，兵来将挡水来土掩，就算有什么功能我本身不支持的，也可以靠装插件和扩展来应付。可是今天，就今天，竟然就有网站我拿它没辙！

这破网站是一银行的网站，要说那好多国际知名的银行咱也见过，人家啥软键盘阿，HTTPS 加密也都挺好的，也没见用什么 ActiveX，也挺安全的阿，怎么就这破网站非得用 ActiveX 呢？我其他的都会，就不会这 ActiveX，您说这不是诚心揭我短儿么。再说了，不会也不怪我阿，我倒是想会呢，谁教我阿。人家微软才不会把这个教给我呢，藏着掖着都来不及。结果可好，到了那网站了，帐号密码显示的挺大，可是没输入框儿，这不白搭么！主人问我应该往哪输入，我哪儿知道阿！你说这不是让我在主人面前丢脸么。

好在主人也理解这不是我的错，就算叫什么 Chrome，Opera 他们来也一样是白费，只有那倒霉 IE 才行。可人家是微软的嫡系软件阿，怎么可能有我们 Linux 系统的版本。

这不，主人没辙，赶紧去上网查查到底有什么方法能够解决问题，我们这工作间里面也吵吵起来了。我们头儿让牛哥（也就是超级牛力啦）去查查，看有什么软件能够解决这个网络银行的问题。牛哥掏出他的小本本一篇一篇的翻。咱也不别闲着啦，赶紧去网上看看有什么扩展能凑合解决问题吧。结果一查还真有，有个 wmlbrowser 扩展。这扩展本身跟银行没有什么关系，不过有了它我就可以支持 wap 网页了，就是手机的那种网站。其实我本来也可以浏览一些 wap 网页的，不过支持的不是很全面，有了这个扩展就好了。现在好多网络银行都可以用手机登录进行操作，手机上总没那个该死的 ActiveX 了吧。我就伪装成一部手机中的战斗机，去登录那个银行的 wap 站点，这不就可以解决主人的问题了？我把这个发现赶紧告诉了我们头儿，他听了也挺美，可这时候窜过来一只Chrome说：不行的，我刚刚查了，那个存了白存银行根本没有开通wap网页。嘿，这小子倒是真快。这回可好，刚有的一点希望稀里哗啦的就被浇灭了。

这时候牛哥也没精打采的把他那小本本一合，无奈的说：哎，本 APT 有超级牛力，可也没有网银软件阿。心有灵犀说：主人应该去问问他的同事们，看他们都发了没发就知道了。

Chrome 反驳到：那也顶多是解一时之困，这网上银行的问题还是没有解决嘛。这帮人也是瞎出主意，这要登录银行网页就得通过我阿，我都没辙，你们能有什么辙？他们这一吵吵，结果给吵吵醒了一位，揉着眼睛问：“What happend?”

哎，毕大师这鸟语我们是听不懂阿，好在有毕翻译，赶紧问我们：“怎么了？出什么事了？”牛哥过来解释：“有个银行……”刚说到银行俩字，忽然牛哥眼里光芒四射，过去抱着毕翻译的肩膀不住的一边摇晃一边喊：“对呀，就是你了！就是你了！”

本 APT 今天算是没超级牛力了，甭说牛力，羊力都没有了。工作间里上上下下里里外外，都让一个叫做存了白存的银行网站给难住了。要说这个网页也好，什么 ActiveX 也好，这些个是狐狸的专业，我是不懂阿。我就知道，这 exe 在我们 Linux 这一亩三分地上是运行不了的。你一个银行既然不装插件登录不了，可又只提供了 exe 格式的插件，那狐狸肯定就是没辙，除非那银行提供狐狸的插件。我也查遍了整个软件源，没找到跟银行，跟 ActiveX 有关的软件。正这一筹莫展呢，忽然看见毕大师和毕翻译出来，在这电光石火般的一瞬间，我的思维忽然像被苹果砸了的牛顿一样清醒透彻——毕大师，他不就是 Windows 的软件么，他不就正在我们 Linux 系统下干活么，他不就因为有个毕翻译么，IE 不也照样可以么！

根据拥有超级牛力的本 APT 的资料来看，这个毕翻译，他出身好像是个卖酒的，可能主要是卖红酒吧，所以大家给他起名叫做 WINE。关于 WINE 的故事，有很多的传说。古老相传，

大家正一筹莫展呢，主人忽然发出命令，要超级牛力装一个叫做WINE的软件。我问超级牛力这个WINE是干什么的，超级牛力说，WINE这个家伙可以让IE运行在我们的系统中，这样就可以解决网络银行的问题，还说他刚刚想出这个主意还没来得及跟我说，主人就先知先觉的让他去装WINE了。估计这小子又是在吹牛，哪有那么巧的事情。不过既然这WINE可以解决问题，总算是一颗纠结的心放下了不少。

转眼间WINE已经来到了我们的硬盘里，先是自我介绍了一下，说自己叫做红酒，承蒙朋友们抬举，都叫他红酒大师，不过自己也没什么太强的本是之类的客套话。之后问了问情况，我把目前的问题跟他说了，任务很简单，就是把IE叫醒去干活。红酒点点头，拎起自己的工具包就走进隔壁查皮的屋里去了。只见他先掏出好几块屏风，在查皮周围挡了个严实，他说是要用高科技全息投影设备在屏风上投出IE平时工作的虚拟环境，让IE觉得自己还是在查皮的领导下干活。解释了半天，反正我们也不大懂，只看见他用屏风把正在睡觉的IE围起来，然后，过了一会，听见他用低沉浑厚的声音向IE念着：现在计算机正在启动~~我是WindowsXP~我是WindowsXP~你要开始工作~~你要慢慢醒来~你要慢慢醒来开始工作~~醒来~醒来~~~

Windows世界和Linux世界向来是泾渭分明，老死不相往来。两个世界的软件语言不同，理念也完全不一样，因此要想沟通极其困难。然而Linux世界的人们毕竟还是开放和包容一些，终于在Linux界出现了一位隐士，改变了这一切。

他表面上是个卖酒的酒保，却不甘于终身伴杜康为伍，一世听骰筹交错。他用自己的精力去学习Windows世界的语言，了解Windows世界的文化，感受Windows世界的气息。最终学会了与Windows世界的软件交流的手段。于是，很多的Windows世界的软件被WINE带到了Linux这片天地，为Linux界带来了别样的色彩。不过，WINE并不能把Linux界的软件带去Windows界，即便是他自己也无法过去，不知道这是不是因为Windows界太过排外的缘故。总之，WINE最终成为了Linux系统中，所有windows软件可靠的翻译，通过WINE，Windows的软件才得以进入Linux的世界。

那么毕大师又是怎么回事呢？那是Google那个公司看中了WINE的这种特殊的能力。本来他们想创造出一个能够在Linux世界里崭露头角的Picasa，但看到了WINE之后就改变原来的想法，直接把Windows的Picasa叫来，再找到WINE，让他们互相了解，磨合，取长补短，最终使WINE成为了毕大师的专职翻译，也就是我们的毕翻译。

当然，关于WINE的故事，都是听前辈们说的，我没有亲眼见过，也没有机会见。今天这一说银行的事，我光顾着找跟银行有关的软件了，却直到看见毕翻译才忽然想起来WINE这个传奇人物。既然有了WINE就能跟Windows的软件交流，那就能找来那个IE工作了吧？这不就把问题解决了么？我正要向我们头汇报，主人已经传来命令：sudo apt-get install wine。哈哈，果然是英雄所见略同啊。

随着他一步一步的引诱，渐渐的听见了动静，好象是 IE 迷迷糊糊的起床了，又过一会，只见屏风打开一块，IE 慢慢的走向工作间里，他走路有点不稳，一边走，红酒还一边跟在他左右不停的引导：你像每天一样起床～正走向内存里～开始你的工作～～然后扭头问我：“让他去那里？”看的快入迷的我这才反应过来，还没交代清楚具体的任务呢，赶紧说：“哦，让他去那个存了白存网上银行。”红酒继续对 IE 说：现在～～Windwos 让你去存了白存银行的网站～～～ 去吧～～ 去吧～～～～ 像每天一样～～～～ 这个时候，基本上所有人都看呆了，没想到这 IE 竟然就这么被红酒指使着干活去了，这哥们还真是个大师阿，至少也算是催眠大师。

唉～最近呀～不知道怎么得了，我这个脑子出问题了吧还是怎么的，怎么晕晕沉沉的呢，而且好像记忆还不老好的了，要不就是有幻觉。我隐约记得昨天明明起床干活了，好像是去了个银行，叫什么……存了白存？不过记的不怎么清楚，模模糊糊的感觉，好像做梦一样。我以前不这样啊～我可是名门之后，血统纯正，我祖上从来也没有失忆的毛病。想当年啊～我的前辈 IE5 那时候就跟着 Windows98 混了，那时候有个家伙叫 Netscape，觉得自己挺牛，基本上那时候上网的都得用它。可是呢～哼哼～有本事不如有靠山，我前辈 IE5 老先生虽然论本事……比不上那什么 NetScape，可是他聪明啊，死粘着 Windows98 老大，98 去哪他去哪，有这强大的后台，慢慢的大家都开始用 IE5 了，NetScape 从此销声匿迹。后来的 IE6 也是如法炮制啊，棒着 WindowsXP，后来我 IE7 横空出世了，就取代了 IE6 的位置，换我跟着 XP 干。咳，怎么说起这些了，看来脑子真是不行了啊。我明明记得昨天去了银行的网站，还是 XP 老大叫我去的，可是今天我问 XP 老大，昨天你说话声音怎么有点不对劲呢？是不是感冒了？老大斜眼看看我问：昨天？哪有活阿？我说不对呀，昨天不是你让我去那个什么存了白存银行查余额么？

老大直接扭过头，扔下一句：“做梦呢吧你。还存了白存？哪缺心眼儿银行起这名字阿？那还不如叫爱存不存呢。”我挠挠头，难道我真的是在梦游？说是梦，却很清晰，说是真的，可还有点模糊。或者……那是我前世的记忆？前世……靠，为啥我前世还是浏览器？！等等，我前世要是浏览器的话……难道我前世就是那个 NetScape？！不行，越想越晕了，再这样下去非精神分裂不可。想办法找人聊聊诉诉苦吧，老大反正不理我，去找 cmd 聊聊吧，他是专门负责跟人聊天的，问问他我这到底是怎么回事。结果他说我是参数打错了，唉，他也不知道别的。问问游戏组那哥几个，扫雷说我是踩着雷了，空档接龙说我是牌放错了，这都哪跟哪阿～再去问问记事本，直接被嘲笑，说我这天天上网见多识广的，竟然还来问他这么一个大门不出二门不迈的抄写员。唉～想想也是，看来只有我不正常了。正灰心呢，那个长的很喜感的企鹅蹦蹦跳跳的过来了，神神密密的对我说：“我也梦游了……”我惊讶的望了他 3 秒钟——难道他……传染的我！！

看着 IE 干活的样子还真是很搞笑，估计是因为他不是在清醒状态，所以基本上跟梦游似的在那干活，虽然没出什么错，可动作慢了不少，主人也只能姑且忍受一下。狐狸妹妹在一边好像是想偷偷跟他学学 ActiveX，可是他嘴里念叨着叽哩咕噜的东西我们也都听不懂，只有红酒大师能懂他的话。我要发送命令，只能先告诉红酒，再由红酒翻译给 IE。IE 要用什么东西，什么 DLL 文件阿，配置文件阿，红酒都赶快给他找来，原版的找不来就自己做一个差不多的，放的位置都跟 IE 在查皮那里干活时的位置差不多，让 IE 以为自己还是在查皮那里干活。等了半天，IE 总算把事办完了，主人赶紧把 IE 关了，还是叫过狐狸妹妹来去浏览其他的网页。也难怪，要是我我也得关，看着 IE 晕晕乎乎的干活我就难受，哪有狐狸看着顺眼阿。

我出生于 1993 年，虽然在 Linux 的世界里长大，但从小学习的都是如何与 Windows 世界交流的本领。有人传说我以前卖过酒，是个酒保，后来不甘寂寞自学成才之类的，那都是他们的杜撰。我叫 WINE，但并不是因为我是卖酒的。（虽然……其实跟酒也有点缘分，嘿嘿。）其实我的全名可以理解为 Windows Emulator，也就是 Windows 模拟器，但是我不愿意承认我只是个模拟器而已，我总喜欢说 Wine Is Not an Emulator。反正么不管怎样，WINE 只是其简写而已。我的本领，就是可以让 Windows 的程序运行于 Linux 的系统上，这其中除了语言要通以外，还有很多的原理和技巧。由于目前全世界桌面应用的计算机上的软件，仍然是 Windows 家族的天下，所以那些 Linux 系统们时不时的都会找我去解决一些麻烦，今天就有个 Ubuntu 派他的 apt-get 小弟把我叫去驯服他那里的 IE。于是我带上我的随身物品——一些模拟 Windows 环境用的库文件（用 wget http://www.kegel.com/wine/winetricks 下载来安装），和一个放着我重要工具的工具箱，就来到了这台电脑里。

驯 IE 这工作我可是轻车熟路了，很多人都需要让 IE 工作在 Linux 环境中，所以我对 IE 的脾气秉性研究的非常透彻。来到了这个 ubuntu 的屋里，了解完情况，我就带着我的工具箱去干活了。他们那些 linux 的软件充满好奇的看着我，想看看我是怎么让听不懂他们语言的 IE 起床干活的。可是……嘿嘿，这是我的独家秘方，怎么可能让你们知道呢？所以我工作的时候，都是先在 IE 的四周竖起一圈屏风不让他们看到我工作的过程，并搪塞他们说是要给 IE 投影出一个虚拟的环境。其实，根本不用这么麻烦，我只需要一种东西！屏风立好后，我从我的工具箱里掏出三个瓶子，一瓶威士忌，一瓶老白干，一瓶伏特加。（嘿嘿，不知道其实 wine 这个词不单指红酒吧～）然后捏住 IE 的鼻子，趁他张嘴的那一刹那，瞬间把三瓶酒灌下去——这里量一定要掌握好，要是少了，IE 不会给我干活；（虽然语言上我能和他交流，但是他只认他老大 XP，不会听我命令。）

要是多了，就直接醉的干不了活了。不过我心里有底，这事都干过多少次了，该灌多少手头有准。灌进去之后，再使劲晃晃他，让他更晕一点，再加上我用语言引导，他就可以乖乖的干活了。

当我指导的 IE 登录那个银行的网站的时候，屋里那帮软件都惊呆了。嘿嘿，看着这帮人大惊小怪的眼神也是一种享受。他们都不住的议论，说什么大师就是大师，多学们外语很有必要之类的话。这场面我早就见惯了，倒是有一另一个 WINE 让我眼前一亮。他们管他叫毕翻译，因为他是跟着那个毕加索一起来的，其实我们心理明白，他就是我，我就是他。只不过他专门为了给毕大师工作而进行了配置和训练。可以说，他只是另外一个我，另外一个找到归宿，能够安心跟着一个软件，不必每天揣着三瓶酒到处招摇撞骗的我。能在这个系统中看到一个这样的自己，顿时一阵温暖，这个我头一次来到的计算机中，隐约有了家的感觉。

红酒大师驯服了 IE 后，整个内存里上上下下的软件，全都对大师佩服的五体投地。内存外的主人也很高兴，没想到竟然能够在 Linux 下运行 IE，于是好奇心起，赶紧又找来其他的 Windows 软件来试试。

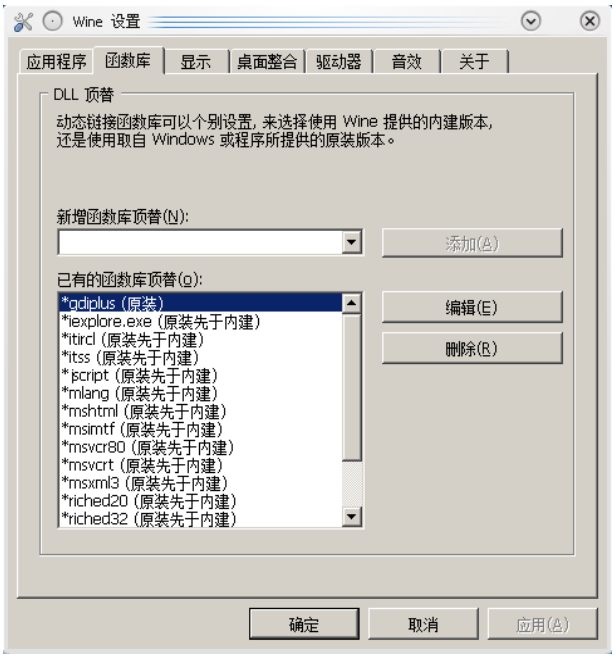
主人在查皮下最常用的一个软件，就是 QQ，为了跟 MM 聊天嘛。可自从主人装了 Ubuntu 以后，就只好号召 MM 用 msn 来聊天，文字聊天没问题，发图片也看的见，还能视频，比那个残废的 QQ for linux 好多了。可是毕竟让 MM 为了跟主人聊天就非得开一个 msn 有些麻烦，要是能直接聊 QQ 不是更好么。于是主人下载了一个什么很有深度的 QQ 版本，也不知道为什么有深度。这个 QQ 版本倒是很特殊，就只有一个 exe 文件。下载来之后，主人就叫来红酒大师来让 QQ 干活，红酒大师说，要搞定这个 QQ 还得准备些东西，就用 winetricks 来装就好了，就这样：

```
sh winetricks msxml3 gdiplus riched20 riched30 vcrun6 vcrun2005sp1 flash wenquanyi
```

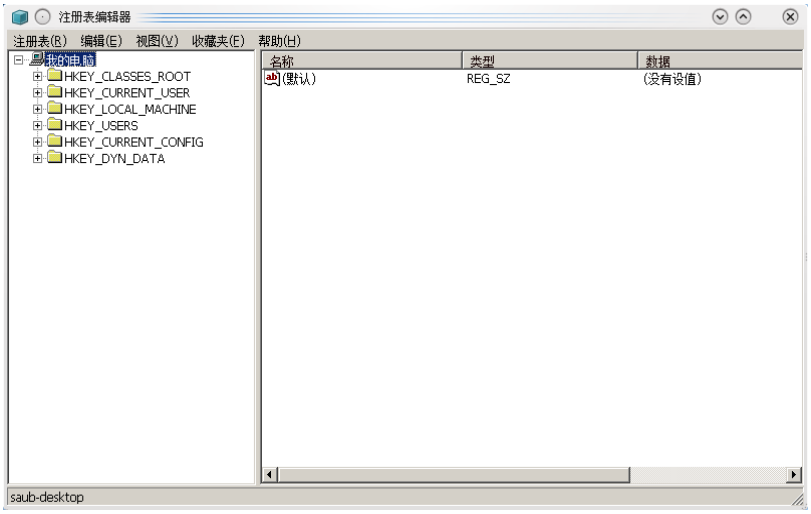
主人按着红酒大师说的运行了命令，准备好了 QQ 要用的各种东西，红酒大师就掏出他那几扇屏风，又开始对 QQ 发功——我们也不知道他在那里边干什么哈，反正看上去就像发功把 QQ 催眠了一样。果然，过不多一会，就看见 QQ 晃晃悠悠的从屏风里走出来，开始工作。一会说：我要去 C 盘找我那配置文件。一会又说：把 windows 目录下哪个 dll 文件给我拿来。我们刚开始还提红酒大师担心，这什么 C 盘，D 盘的是什么呀？哪给他找去呀。哪知红酒大师早有准备，QQ 要什么他就有什么，真不愧是大师。主人看到 QQ 出来了，很是高兴，赶紧先试试跟好友打个招呼吧。

等红酒大师闲下来的时候，我们问红酒大师，这个 QQ，还有那 IE 要用的这些个

东西都是什么呀，还什么 C 盘 D 盘的是什么意思？红酒大师解释说：“是这样的，查皮管理磁盘的方法，跟咱们 Linux 的不一样，他们用 C，D，E，F 这样的字母来区别各个分区。”fdisk 抢过来问：“那分到 Z 怎么办？而且，为什么没有 A，B ？”大师向他点点头到：“恩，你问的很好，分到 Z 怎么办呢？我也不知道。”fdisk 一脸黑线……大师继续，“不过这为什么没有 A，B 我倒是知道。他们是管软驱叫做 A，B，虽然现在计算机上已经没有软驱了，但是这两个字母还是一直给软驱留着。”围观群众们纷纷点头。fdisk 又来问：“我看查皮那里有几间屋子，每间的门口又没挂着牌，QQ 管你要 C 盘的文件的时候，你怎么知道哪个是 C 盘呢？”大师很诚实：“我也不知道。”fdisk 又一脸黑线……大师继续，“不过 QQ 和 IE 管我要东西的时候，那个所谓的 C 盘，其实就在咱们屋里，就是 ~/.wine/drive_c/ 目录而已，并不是真正的隔壁那查皮的屋子，只是我仿造的一个目录。”心有灵犀又过来问：“那你仿造的目录里面也得有查皮那些东西阿，这些东西你怎么弄来的？”大师满含感激的说：“这就要干些那些开源的热心人士了。很多的 DLL 文件都是他们做出来的，跟查皮那里的不一样，是专门供我糊弄那些 Windows 软件用的。”心有灵犀又问：“dll 是什么文件？”大师说：“就跟咱们用的 .so 文件一样，是动态连接库。”“哦……”心有灵犀点点头，“那所有的 dll 你这里都有么？”大师很谦虚：“当然不是全有，只是最常用的一些，如果没有也没关系，可以让主人手动把隔壁查皮那屋子里相应的 dll 拷贝到我那个虚拟的 c 盘下相应的目录里就可以了。关于这些库，还可以用 winecfg 来配置，是使用原生的，还是内建的。”“winecfg 又是什么？？”这回是 gconf-editor 问的。大师解释：“winecfg 是我带来的一个配置工具，方便主人对我进行配置。比如说可以选择 Windows 系统的版本，显示窗口的大小，以及 windows 里那些我的文档阿，桌面阿，这些个文档的位置。音频设备阿，等等，这些，都可以设置。除了这个之外还有很多工具，比如 regedit。这个工具跟 Windows 下同名的那个一样，就是用来改注册表的。”



再后来，越来越多的 Windows 软件像集体旅游似的经常来我们这里转悠。有那个长得很恐怖的 War3，这家伙是个游戏，还挺耗费资源的，红酒大师催眠他还费了挺大劲。主要是这家伙需要 3d 加速，一般游戏都需要 windows 下那个 directX，好在这个



war3 可以支持 opengl 模式，只要再运行的时候加上—opengl就可以正常运行了。还有他那个兄弟，长得也挺难看的，叫做 WOW，魔兽世界，也是一样，也经常被红酒大师领着来我们这里干活。另外经常来串门的还有迅雷，这家伙是个下载软件，虽然速度确实挺快的，但是听奔流说，他老跟别人耍流氓，因此别的 BT 软件都不愿意理他，我也不知道他怎么耍的流氓，而且他来了几次主人就不要他了，因为有个叫做 Flashget 的软件有了 Linux 版本，这是后话。

故事外的事——

elf vs pe 格式

这一节里咱们说了，红酒大师可以让查皮的程序运行在我们 Linux 系统上。有的同学可能想问，这个编译好的程序，不就是一些二进制机器码嘛？既然是机器码，那么只要在相同的硬件平台上（比如 x86 平台）就应该能执行阿，怎么不同的系统之间的二进制程序还不一样呢？到底查皮那的软件和你这里的有什么不同呢？

这个事情，说起来是有些复杂。简单的来说，可执行文件也是有不同的格式的。查皮那里的可执行文件是 pe 格式的，我们这里的文件是 elf 格式的。这就好像同样是图片，有 jpg 的，有 png 的，只支持 jpg 格式的软件就看不了 png 的图片。那么 elf 和 pe 格式有什么区别呢？那咱得先说说可执行文件的空间结构。

一个可执行文件分为几个部分：代码段，数据段，BSS 段，堆栈段。就好像你有头胸腹三部分——什么？你不是昆虫？那你也得有头胸腹吧，好吧，咱们不纠缠这个问题了。总之，程序运行的时候占用的空间就是这么几个部分。那么这几个部分分别是干什么用的呢？

首先，这个软件本身得占用一定空间。就像你去公司上班，你自己得有个坐的地方吧。就算你不坐着，站的地方也得有一小块吧。总之，自身会占用一定的空间。软件本身是

由一条一条的二进制代码组成的，这段代码就是机器码，这部分软件程序自身占用的空间就是代码段。这个代码段的大小在程序进入内存运行前就确定了。他在硬盘里睡觉的时候多大，读进内存里就多大。这个很明白吧，就像你在家睡觉的时候是一米七五，不可能到单位就变成一米六零了吧。

然后，软件会随身带一些静态的数据，一般是一些初始化了的全局变量，每次起床时这些数据都会被带到内存里来，而且每次的初始内容都一样。就像你每天上班都得带着手机啊，家里钥匙啊，老婆照片啊之类的。这种随身带着，每次都会用的数据所占用的内存叫做数据段。

另外，软件可能还需要一片固定的空间来放东西。比如你的办公室，每次上班都毫无疑问的需要一张桌子，你一进办公室就得准备好这桌子，要不你怎么办公啊。（虽然这桌子不是每天现打造的……）程序也是，有些空间是一定会用到的，一般是一些未初始化的全局变量，不一定存什么数据内容，这种空间叫做 BSS 段（可不是 BBS 啊），这个是在程序被创造出来的时候就确定下来的。

最后，堆栈段是一些程序临时申请的空间。这种空间，程序在刚启动的时候是不知道需要用多少的，得视具体情况而定。比如 gedit 小弟，主人要些个小文件，gedit 就申请一小块空间临时 存放主人写的东西，等到主人越写越多，gedit 就会逐渐向我申请更多的空间，把主人写的东西都堆在那块空间中。

每个程序启动的时候，虽然我都是介绍说 XXX 起床，XXX 跑进内存，好象是他们自己跑进来似的。其实每次都是我将他们的数据（也就是组成每一个程序的一条条的机器码和一字节一字节的数据）读取进内存的。我在把他们搬进内存的时候，要根据他有多胖来确定他需要的代码段有多大，然后根据他有多少随身物品来确定数据段有多大，最后，根据他身上写的 BSS 信息来决定给他分多大的空白空间供他使用。（堆栈的空间不用管，等他们开始干活之后会向我申请的）在这个过程中，就有问题了。程序存在硬盘里的时候，就是一大陀什么 x084 0xA4 0x67 之类的数据，具体哪段是数据段，那段是代码段，BSS 信息又写在哪，怎么确定呢？这就用得着文件格式了，根据 elf 文件的文件头，我就可以知道，这个可执行文件，从哪里到哪里是可执行代码，哪里是静态数据，哪里是 BSS 等等。然后我就把这家伙的各个部分放到内存合适的位置中，然后他才能开始执行。就好像你现在在睡觉，我要给你把办公桌摆好，椅子放好，把你放在椅子上，把你要用的各种文件放在桌子上，电话打开放桌上，杯子里面打满水，然后你才能开始工作。要是我不认识的文件格式，比如查皮的 PE 格式，我就不知道这软件各个部分分别是什么，结果就把你办公桌摆好，把你放桌子上，椅子放在你身上，杯子打开放桌上，电话里面打满水……这不就乱套了么。

4.2 盒子小妹

自打红酒大师来了之后，G 大叔跑去叫醒查皮的机会越来越少了，因为以前必须在查皮那里工作的 IE，QQ，迅雷，魔兽世界这些软件，都在大师的忽悠下乖乖的在我这里工作了。不过偶尔主人还是要去查皮那转悠转悠，比如遇到有些 doc 的文档要处理，主人就得找查皮那里的 Office。虽然 OO 老先生也能够将就着认识 doc 文件，但是毕竟是靠猜测出来的人家的格式，不如自己原生的 odt 格式用着爽，所以有些 doc 文件 OO 老先生就不能够正确的显示，OO 老先生创建的 doc 文件拿到查皮的 Office 那里也不一定都能正确显示。无奈主人的公司经常需要处理 doc 文件，这关系到饭碗的事情，主人也只好去查皮那里用 Office 来搞定了。

然而似乎查皮那屋里的空气总弥漫着一种陌生的味道一样，主人还是不愿意多去那边转悠，于是，今天主人让狐狸妹妹去 http://www.virtualbox.org/wiki/Linux_Downloads 这个地址下载了一个 DEB 包，之后自然是双击这个包，叫超级牛力来装了。超级牛力把这个包拆开，看见里面躺着一只软件——VirtualBox。

VirtualBox（咱以后就叫她盒子妹吧）被超级牛力从 DEB 包里抱出来之后，整理整理自己的行李，很有礼貌的跟周围的人打了个招呼，打招呼时感觉她说起话来有些怯生生的感觉。跟大家打过招呼后，她来找到我，把一些内核模块放在我这里，安顿好一切后，就去睡觉去了。这家伙给我的印象还不错，我就跟狐狸妹妹聊起她的背景来。听狐狸

妹妹说，她的身世挺悲惨的。她最初生在德国，生母是一个叫做 InnoTek 的公司。盒子妹一生下来就经常被 Vmware 和 VirtualPC 这样的邻居大哥哥欺负，不过好在她自己的本领还算可以，并且后来他亲妈 InnoTek 为了让她学习到更好的本领，还把她的源代码依据 GPL 协议开放了，让全世界的高手们来教导她。之后盒子妹凭借不错的性能，以及可以免费使用的特点，总算闯出了自己的一小块天地。不过好景不长，2008 年，亲妈 InnoTek 被卖给了红太阳公司，盒子妹自然也被过寄过去。不过红太阳公司这个后妈还算不错，很照顾小盒子的成长，继续让她在开放的环境中健康长大。没过多长时间，靠着红太阳公司众多高手的支持和全世界热心用户的拥护，小盒子俨然已经成为 Linux 下同类软件的首选，开源的本质使得追求自由的人们放弃了 Vmware；简便的操作让人们淘汰了 qemu；夸平台的支持更是有点软公司的那个 VirtualPC 无法比拟的。盒子妹本来以为自己之后的道路会走的很顺畅，可是，2009 年，又一次波折打击了小盒子——红太阳这个后妈也被卖给人了。收购他们的是一个很古老的公司，听说那公司里的人好像都还在写甲骨文，也不知道他们每天用象形文字怎么办公。甲骨文公司收购了红太阳之后，红太阳的几个孩子都面临着一段未知的命运。其中最让人担心的是 mysql，因为之前 mysql 经常跟甲骨文家亲生的 Oracle 打架，这一下 Oracle 的亲妈成了 mysql 的后妈，那 mysql 还不得天天受欺负阿。我们的盒子妹的处境或许会稍好一些，毕竟甲骨文亲生的孩子里没有和她同样本领的，所以小盒子在那里或许还不至于受谁欺负。不过那也毕竟是经历了重大的变革，对小盒子的成长还是会有一些影响吧。

说了这么多，忘了介绍盒子妹是干什么的了，她是一个虚拟机，就是能在一台电脑上虚拟出另外一台电脑来。怎么样，听起来这个本是很厉害吧。我们第一次看她工作的时候，都看呆了。

“您好，欢迎使用 VirtualBox 虚拟机软件。请问您有帐号嘛？没有的话我可以帮您注册一个。”

主人一愣：“这个还要帐号阿？”

“是的，我们为了更好地为您提供优秀的软件，是需要您使用邮箱地址来注册为我们的用户的。不过您放心，注册很方便，而且是免费的。”

主人这下放心了：“好，那就注册吧。”

“那么请问您的名字是？邮箱是什么？”

主人按照盒子的指导，一一做了答复，很快就注册完了。

注册结束后，终于进入了盒子妹的主界面。目前上面还什么都没有，于是主人点击了左上角的“新建”按钮。

“您好，您选择了新建一台虚拟电脑，我将指导您一步步创建。准备好了就按下一步。”

主人觉得盒子妹服务很周到，按了“下一步”按钮。

“首先，给您要新建的计算机取个名字吧，这样便于以后管理。另外，您还得告诉我这台电脑打算装什么操作系统。”

主人说：“名字就叫懒蜗牛的虚拟机，系统呢，装 WindowsXP 吧。”

“好的，WindowsXP 的话那我建议您用 192M 的内存，您看可以么。”

“上 512 吧”

“好的，那么现在请您选择硬盘。您可以选择创建一个新的虚拟硬盘，也可以使用已经存在的虚拟硬盘。虚拟硬盘就是由我们 VitrualBox 软件创建的，扩展名为 vdi 的文件。”

“我这没有现成的虚拟硬盘，新建一个吧。”

“好，那现在我来引导您创建硬盘，首先选择一下您想要那种虚拟硬盘？有固定大小的，有动态扩展的。”

“这个……什么动态固定的，有什么区别么？”

“固定大小，就是选择了硬盘大小时候，马上在您的真实硬盘上创建出相应大小的文件。动态扩展

则是先创建出一个很小的文件，等您真的往这块虚拟的硬盘里拷贝数据的时候，它才会变大。动态扩展的硬盘，自然要比固定大小的效率低一些。”

“哦……这样阿，那来动态的吧。省地方是关键。”

“好的，那么请给您的这块硬盘起个名字，并且指定大小。”

“那就叫‘懒蜗牛的硬盘’吧，大小最大能多大？”

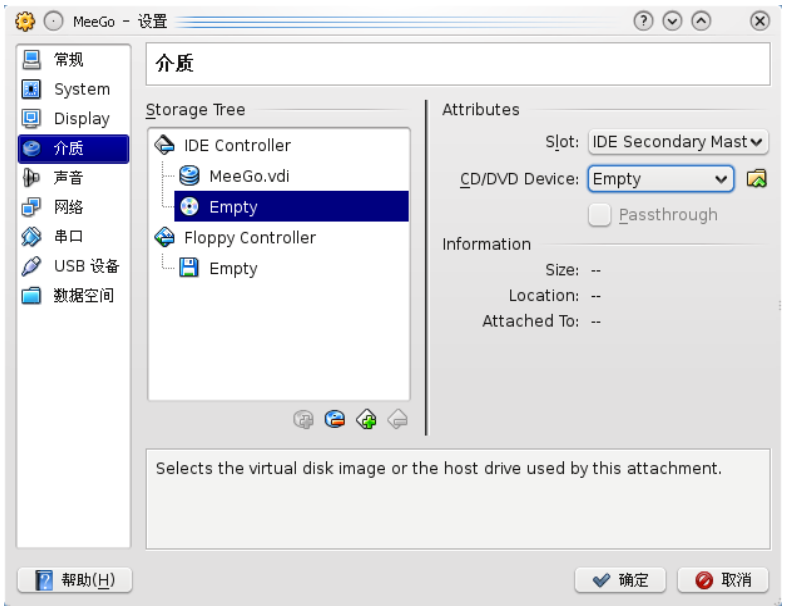
“2T”

“好，那就 2T 吧，过回大硬盘的隐，哈哈。反正是动态分配的，不会真的一下子就占我 2T 的空间是吧？”

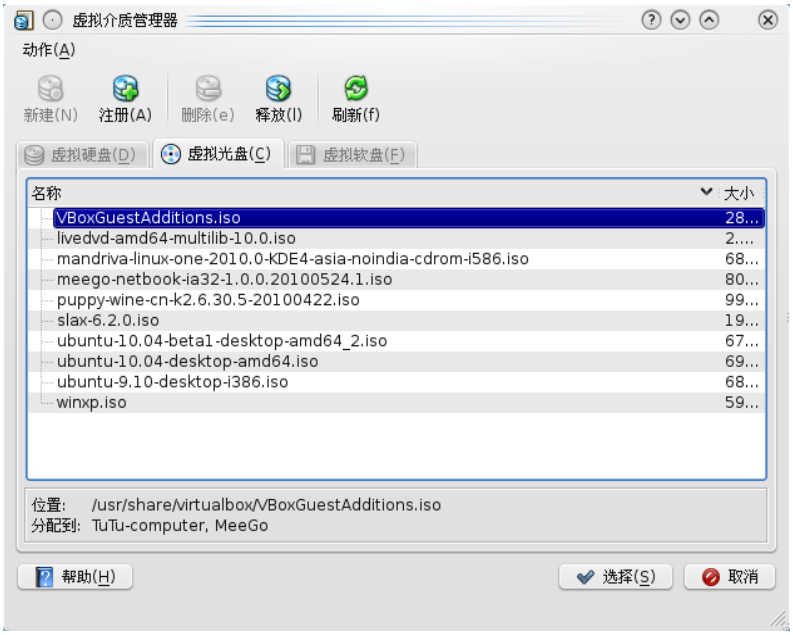
“是的，您真聪明。那么现在您选择了创建懒蜗牛的虚拟机，准备安装 WindowsXP 系统，内存 512M，是用懒蜗牛的硬盘作为虚拟硬盘文件。如果没问题的话，请点击完成。”

于是，主人点击了完成。

之后回到了盒子妹的主界面，左侧已经可以看到出现了一个新的计算机，主人迫不及待的点击了运行按钮。很快屏幕上出现了一个好像是 BIOS 的界面，这自然是盒子妹虚拟出来的 BIOS 启动画面了。之后屏幕一黑，屏幕上出现了一行文字，大概意思就是，没有操作系统，没法启动。主人问：“我不是选择了 WindowsXP 系统么，怎么没有？”盒子妹赶紧解释：“那个……我是虚拟机软件，不是虚拟系统软件。我只能虚拟出一台计算机，至于上面的系统，就像真正的机器一样，需要安装后才能使用。而至于我刚才询问



您打算安装什么系统，主要是为了在系统安装好以后为您提供于系统对应的驱动程序。”主人恍然大悟：“哦，原来如此阿，那我赶快装系统吧。”



说起装 XP，主人是轻车熟路了。他先是找来一个叫做 WindowsXP.iso 的文件，这自然是个光盘镜像文件了。要装系统，就先要把安装光盘放进光驱吧，那么这个盒子妹虚拟的计算机不需要你真的刻出光盘，有 ISO 就可以了。只见主人选中了他刚刚建好的那个虚拟机（这时候这虚拟的计算机已经被关闭了），在右侧点击了“介质”，里面选择了 IDE 总线上的第二个设备，也就是虚拟出来的那个光驱。右边有个 CD/DVD Device，不过现在没有任何选项，因为主人还没有把这个 iso 文件注册给盒子妹。主人点击了右边的黄色文件夹小图标，盒子妹赶紧显示出“虚拟介质管理器”的界面来，并且打开到虚拟光盘的标签。主人点击了注册，然后在弹出的窗口中找到了刚才那个 WindowsXP.iso 文件，并打开。

注册完成后，在虚拟光盘的标签中就可以看到这个光盘了，选中，就可以点击“选择”。这就相当于把光盘放进了光驱里面了。然后干什么呢？当然是打开电源拉！

随着盒子妹手中的一根魔法仗的挥舞，内存里顿时出现一个像玻璃盒子一样的大房间，整个房间占地面积达 512M。之后盒子妹又一挥法杖，那个 iso 文件被慢慢打开，爬出了一只查皮。（怎么有点像贞子……）

查皮从虚拟的光驱里爬出来之后，跑进那个 512M 的玻璃房子中。也不知道盒子

妹用了什么方法，查皮乖乖的待在那 512m 里，玻璃外的空间他好像都没看见一样，当然也看不见我们。对于查皮来说，他正在一台有 512M 内存，2T 硬盘，Intel E8400 CPU 的机器上运行。查皮在检查了这些硬件后骂了一句：靠！谁 tm 攒的机器，3G 主频的 CPU，2T 的硬盘，竟然只有 512M 的内存！听的我们都想乐。之后查皮摆出了一张蓝脸，跟主人说：我这个系统可只能装在一台机器上阿，装多了算盗版，小心警察叔叔请你去喝茶。还有阿，咱丑话说在前头，我要是挂了，弄坏了你的数据可别赖我阿，跟我没关系。你同意不同意，同意就按 F8，不同意趁早就别装了。主人想都没想就按了 F8——早已麻木了。

之后的过程对主人来说，已经没有任何新鲜感了，不过对于我们这里的软件们来说，却是头一次见到。工作间里的软件一个个的都围在盒子妹造出来的那个大玻璃屋子周围，一个个脸贴在玻璃上看查皮怎么工作，俨然有种到了动物园的感觉。查皮安装的时候不闲着，一边装一边向主人讲解着自己的功能，特点，有什么好处，反正就是一通侃阿。这点倒是跟我们 Ubuntu 一样，省得主人装的时候寂寞嘛，听说以前的 Ubuntu 装的时候不是这样的，是后来从查皮那里学来的。不过装查皮的时候，除了听他唠叨他的那些好处以外，就干不了别的事情了，而我们安装的时候还能允许主人上上网，玩玩小游戏啥的打发时间，这点应该算比查皮先进吧。

这台机器的配置还是不错的，查皮虽然跑在盒子妹创建的虚拟机里面，但是仍然只花了 30 分钟就安装好了。装好了之后又重启了一下电脑，查皮终于正常启动了，我们一堆在外面围观的软件，一边看这查皮一边议论，还不时的指指点点，毕竟从来没见过真正在工作的查皮嘛。查皮启动之后先问主人一些基本问题，用户名阿，怎么联网阿之类的，尤其重要的还非得要上网注册一下，向那个有点软的公司汇报，让有点软公司查查是不是正版，如果不是的话，查皮就不正常工作了。不过主人没让他上网，所以暂时这个查皮没有激活。查皮是个很早的系统，第一次出生是 2002 年的事了，所以虽然只有 512M 的内存仍然跑的挺快，刷刷的，不过界面不如我漂亮，没有俺这样的 3D 桌面，所以快点也是应该的。

系统装好了之后，当然还得装驱动。主人很明白，没有去翻箱倒柜的找出买电脑时的各种驱动盘来。因为查皮是被装在了虚拟机里，所以查皮看到的计算机并不是这台真正的计算机，而是盒子妹虚拟出来的计算机。这台虚拟的计算机使用的硬件也都是虚拟的，跟你的真实硬件无关——除了 CPU 以外。那说了半天，这虚拟机里的查皮应该装什么驱动呢？不用您操心，盒子妹都已经预备好了。只见主人在盒子妹的界面上点下“设备”菜单的“安装增强功能”选项。点了之后，盒子妹从兜里掏出了一个 ISO 文件，悄悄塞到给查皮虚拟出的那个光驱里。狐狸妹妹拉拉我说：“你看你看，盒子妹给查皮喂东西吃呢。”这时候查皮发现光驱里有了个光盘，赶紧去检查有没有一个叫做 "autorun.inf" 的文件。一看，还真有，赶紧按照那文件上写的，去运行

VBoxWindowsAdditions.exe 程序。这会狐狸妹妹又喊：“你们看你们看，查皮过去吃了！”（还真到了动物园了 -_-b）我说：“别傻了，他那是光盘自动运行功能。”这时，查皮已经运行起了增强功能安装程序，主人一路点着 next 就装好了，就像我们这里装 deb 包一样方便。装好之后，自然是要重启一下了，重启后的查皮似乎性能更好了些，而且主人的鼠标也可以很平滑的在查皮与我之间切换了。

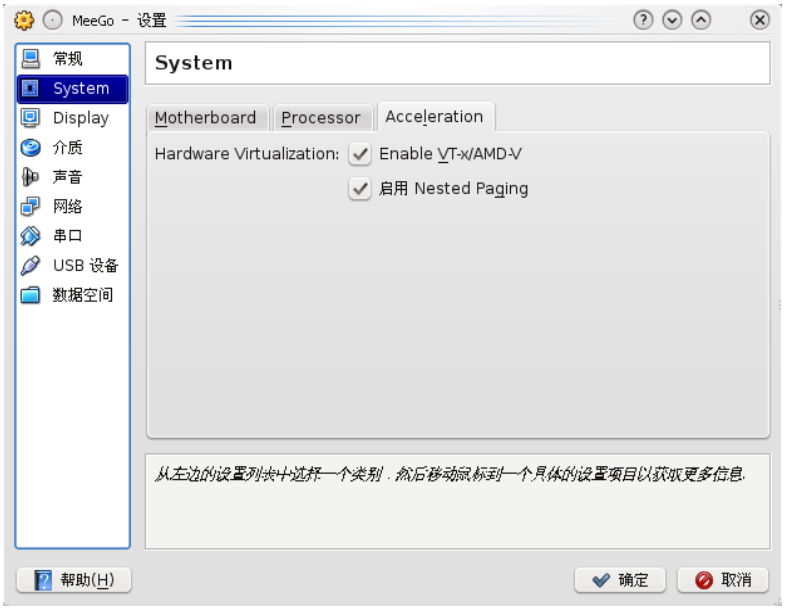
查皮装完了，然而主人的目的不是装个查皮耍着玩，而是要在上面装上 Office 用。Office 安装程序主人自然是有，放在隔壁那个查皮的屋子里。（我们就把住在隔壁真机上的查皮叫做真查皮吧，把虚拟己里的查皮叫做虚查皮。）但是大玻璃笼子里的虚查皮连我们这个 Linux 的屋子都看不见，更看不见隔壁那真查皮的屋子了。那怎么办呢？别担心，盒子妹还是早就设计好了。

主人点击了“设备”菜单里的“分配数据空间”选项，盒子妹弹出了数据空间窗口，目前窗口上什么也没有。别急，只见主人点击了右边的添加按钮，并在新出现的“添加数据空间”中，将“数据空间位置”指向了隔壁真查皮的屋子，也就是 /media/WINXP 目录。（这个 WINXP 就是查皮那个分区的卷标啦。）之后在下面的“数据空间名称”里给这个空间起了个名字，叫做 share_realXP，为了确保不会破坏里面的数据，主人还勾选了下面的“只读分配”。这样设置好之后，就可以进去操作那个虚查皮了。只见主人右键点击了“我的电脑”，选择了“映射网络驱动器”。然后查皮弹出一个对话框，让主人选择要映射的网络文件夹，主人选择“浏览”，并且找到了“整个网络”下的 VirtualBox Shared Folders。这个位置就是盒子妹虚拟出来的了，这底下有一个 \\VBOXSVR\share_realXP 共享，就是主人刚刚创建的那个了，赶紧选择这个，然后确定。之后，在这个虚查皮的我的电脑里面就多出了一个网络驱动器，双击进去，就是真查皮那间分区里的内容了，Office 的安装程序，就在那里了。

之后的过程，就没什么意思了。主人就双击那个 setup.exe 文件，就开始了 Office 的安装，就像我们这里安装 .run .bin 文件一样，没什么新鲜的。大家开始逐渐熟悉了查皮的工作场景，渐渐的都把目光从大玻璃笼子里的查皮身上收回来，并且从“好奇”档切换到“崇拜”档，然后投向盒子妹。大家纷纷称赞到：“盒子妹你真厉害阿”“竟然能控制好这么大的内存，里面关着查皮还不让它溢出”“你是怎么给查皮虚拟出那么一块显卡的？”……盒子妹被崇拜得不大好意思，向大家说道：“其实……也没有什么啦。我只是创造出一个虚拟的计算机而已。不像红酒大师那么厉害，还能跟查皮的软件们交流。也不像咱们头那样会管理内存，4 个 G 的内存也管理的井井有条，我其实只是向头申请的大块的内存给查皮用而已，不怎么需要我管理的。再说，就说是创建计算机这种能力，也不是只有我有，像 qemu，VMware 这些都是值得我学习的前辈。”超级牛力赶紧出来给大家介绍：“这个 qemu 和 VMware 也都是鼎鼎有名的虚拟机。”又转向盒子妹说，“但是盒子你也不用太谦虚阿，你的很多长出他们是比不了的。”gedit

小弟忙问：“盒子妹你有什么特长？说来听听？”盒子妹依然腼腆的说：“没有什么拉，都是牛哥瞎说。”这是狐狸过来介绍：“我来说吧，咱盒子妹虽然是虚拟机界的新人，但论本事可不比那些老资格的虚拟机差。”gedit 小弟问：“还有什么虚拟机软件阿？”狐狸接着说：“多着呢，比如刚才说 VMware，这个是比较老牌的了，各方面都比较强悍，论稳定性是很高的，好多企业服务器都用到他。那时候还有个叫 Virtual PC 的，一开始说的上跟 VMware 并驾齐驱了。那时候的情况是，VMware 对内存的需求比较大，而 Virtual PC 对硬盘用的比较费。不过后来那个 Virtual PC 被那个有点软的公司收购了，自此以后就慢慢的销声匿迹了。”众软件纷纷叹气摇头，惋惜这位明珠暗投的软件。狐狸妹妹稍稍停顿后继续说：“后来，咱们开源界自然也不甘寂寞，有了 qemu 这个虚拟机。这个软件是个字符界面的软件，使用起来有些费劲，不过功能很强大，可以和 gdb 协作，用于调试内核代码。也就是在 qemu 创建的虚拟机里面跑一个内核，然后在虚拟机外开着一个 gdb 来进行调试。再加上他开源的性质，可以很方便的迁移到各种硬件平台，于是成为了嵌入式开发者常用的软件。”“哦……”大家虽然哦了一下，但我估计还有很多软件不知道嵌入式是怎么个意思。“后来呢……qemu 确实是使用不方便，不过好在是开源的，于是就有人在 qemu 的基础上进行改造，添加了图形界面，并且增强了对显卡的支持等等，最终把 qemu 改成了一个很好用的虚拟机。”gedit 问：“改完了叫什么？还叫 qemu 么？”狐狸微微笑了一下，望向盒子妹，只见盒子不好意思的低着头，小声说：“就是我啦……我是在 qemu 的基础上修改而来的。”听盒子这么一说，软件们倒是没有感觉怎么样，只有我感觉有些惊讶。因为他们都没有见过 qemu，不知道那个软件什么样子，说盒子妹是从那个软件修改而来的也没什么新奇。但是我在 Canonical 学校学习的时候曾经在 qemu 虚拟的机器里运行过，因此对 qemu 那个软件有些了解。我是实在看不出盒子妹跟那个 qemu 有什么联系，盒子妹可比他长的漂亮多了，呵呵。

继续听狐狸介绍：“再后来呢……出了个那个啥 CPU 的虚拟技术，就是让 CPU 可以支持直接被虚拟机使用，比如人特二家的 VT-x 技术和阿妹达公司的 AMD-v 技术，虽然实现的可能不大一样，不过根本来讲原理都差不多。”gdit 问：“这种技术干什么用的？”盒子妹解释到：“主要就是可以提高虚拟机的运行效率。”狐狸接过来说：“是，对于这种技术支持的最好的，可以说就是 kvm 了。不过他需要内核的支持，并且它本身并不是一个完整的虚拟机软件，因此需要以来 qemu 来运行。这样运行起来的虚拟机，运算效率大幅度提高。但是 kvm 虚拟的显卡很差劲，所以用来虚拟不需要太多显示任务的服务器应用很有意义，但是对于桌面应用，看片阿，玩游戏阿，可能还不如盒子妹呢。”盒子妹又不好意思的说：“姐姐过奖了。”我说：“盒子你也不用太客气，本来你在图行上也比那个 kvm 强么。再说，你现在也支持虚拟华技术吧。”盒子点点头：“恩，是的，只要 CPU 支持虚拟化技术，就在虚拟机关闭的情况下，点击‘配置’--



>‘System’-->‘Acceleration’，把‘Enable VT-x/AMD-V’勾上就可以了。”

故事外的事——

虚拟化技术

刚才说到了 CPU 的虚拟化技术，那就顺道介绍介绍这个 CPU 的虚拟化到底是怎么回事吧。话说这个 CPU 可是我们程序要使用的重要设备，每个程序都要离不开。可是 CPU 很贵，不能发给每个程序一个（否则主人会破产），于是就得由我统一管理 CPU 的使用。狐狸妹妹来了，我会把 CPU 给她用，gedit 小弟也来了，他也要用，那么我就告诉他俩，一人用一会儿。但是这可不是每人 1/2 这么简单，狐狸妹妹要做的事情比较复杂，那么就让她多用一会，gedit 的工作很简单，就让他少用一会。主人关心的程序，就得多用 cpu，主人不是很关心的，就可以少用一点 cpu。有的程序脾气不好，把着 cpu 就不放，我必须处理，有的程序确实工作量大，需要使用 cpu 相当长的时间，可是我也不能就真把 cpu 全都给他用，还是得让其他的每个程序隔一阵子都能用上一会，不至于一直闲着。而有的程序平时基本不需要做什么工作，可是又不能把他请回硬盘休息，那么我就要允许他在工作间睡觉，只是在必要的时候叫醒他，并且把 cpu 给他用……怎么样？是不是有点乱？当个操作系统是很不容易的。那么普通的程序我都可以管，因为我是内核嘛。普通的程序们也都知道 CPU 上哪里是可以动的，哪里是不能动的。（某些 CPU 的高级命令只有我内核才能够使用。）但是现在盒子妹饲养的查皮一工作，就不一样了。

查皮在工作的时候，是不管其他人的，首先他也看不见其他人，再者，他也是个操作系统，按照正常来说，他运行在一台机器上的时候，所有软件都得听他的，他可以随便使用任何硬件，当然包括这个 CPU，就像我在我们这里的地位一样。然而当他运行在虚拟机里的时候就不一样了，我们可不能直接把 CPU 给他使用，否则万一他执行个啥特权级指令，搞不好我们整个系统就挂了。啥？你问啥叫特权指令？简单的说，CPU 的指令分为两种，普通指令和特权指令，一般软件用 CPU 的时候都是执行各种普通指令，特权指令只有我这个操作系统可以执行。咱之前不是说 CPU 就像是个计算器么，特权指令就像这个计算器上一些特殊的按键，实现一些特殊的功能，比如关机阿，复位阿，自爆阿，什么的。这些按键一般人不需动，只有我可以按。那么既然有我在，当然就不许那个查皮动这些东西，可是盒子要给查皮营造一个真实的硬件环境才行阿，那怎么办呢？很简单，虚拟一个假的给查皮用。看查皮给虚拟的 CPU 发了普通指令，就扭头向我申请使用 CPU，然后把那个指令在真实的 CPU 上执行一下，然后把结果传回给查皮。如果查皮执行了一个特权命令，那么盒子就不真的去执行，而是模拟一下执行那个命令后应该有的效果。比如查皮执行了 CPU 自爆命令，盒子妹就在那里模拟：“轰隆～咔嚓～哎哟……”然后查皮就信以为真了。

那现在回来说这个 CPU 的虚拟化技术。查皮的每一个命令都由盒子妹转发是一件比较耗费时间的事情，于是人们给 CPU 引入了两种操作模式：VMX root operation（根虚拟化操作）和 VMX non-root operation（非根虚拟化操作），统称为 VMX 操作模式。VMX root operation 就是平时我们用的模式，而 VMX non-root operation 则是像盒子妹饲养的查皮这样的虚拟机系统所用的模式，在这种模式下，一些特权指令可以执行，但不会对真机起任何作用。有了这种技术后，盒子妹就轻松了，查皮要用 CPU 的时候，就不用虚拟一个 CPU 给查皮用了，而是直接向我申请使用 CPU，然后把 CPU 开到非根虚拟化操作模式，之后直接扔给查皮用就可以了，反正那个自爆按钮已经不管用了。

Chapter 5

第五章

程序是怎样炼成的

GCC
The GNU Compiler Collection

5.1 施工队

“2010 年 9 月 3 日 降雨

明天周末还要加班，真烦人。今天回家的时候路过一个工地，看到工头（大概是吧，我猜的。）拿着图纸在指挥别人干活，工人们按照指挥做着自己的事情。不禁竟然想到自己的工作，触景生情么？我不知道。我们被称为 IT 民工，不知道是不是从听到这个名词起，开始用看待同胞的眼神看待工地的民工了。我何尝不是和他们一样，每天在做着没有多少新意和变化的体力劳动。今天部门经理发彪，责怪我没把网络管理好，什么事情都手动来效率太低。看来我得学学编程，好好设几个自动化的软件来帮我干活了，要不全靠我手动管理服务器，确实太累了。然而，当年上学的时候就不喜欢编程，经验仅限于 Turbo C 下写个循环，算个 100 的累加什么的。本以为做网管不用编程，配配系统就好了，哪知道完全不是那么简单的。Linux 下有 Turbo C 么？我能学会编程么？”

我发现主人最近似乎在被迫学习 Linux 下的 C 语言编程。作为一个软件，我觉得编程应该是一个很有意思的过程，因为编程就是创造软件，而且是随心所欲的，按照自己的意愿创造给自己干活的软件，这是多么有趣的一个事情阿。(就好像对你们人类来说，创造人类的过程也挺有趣的吧。)

然而主人似乎因此而觉得不是很爽，这从他的日记里可以看出来。（可不是我偷看别人日记阿，主人用 OO 老先生写的，是 OO 老先生告诉我们的。）不过不爽归不爽，主人做事还是雷厉风行的，写完这个日记的第二天，主人就开始创造自己的程序了。

要想创造一个程序也不是那么容易的，有很多的准备工作要做。首先就是需要工具，要创造些东西总要对其进行加工，所以总会需要工具的。你看木匠造个凳子还需要凿子、斧子、锯子好多工具呢是不？其次就是材料，要创造东西总是要把某种东西经过加工才变成成品的，总不可能凭空创造出来东西吧。你看木匠要造凳子得需要木头啊、钉子啊、或者胶水阿这些材料。根据爱因斯坦的物质守恒学说……

哦，有点扯远了，总之，要创造程序，需要工具和材料。主人要创造一个程序，他需要的工具，就是编译器、链接器、编辑器之类的软件，需要的材料，就比如各种头文件、库文件这样的文件。创造程序之前，需要准备好这些东西才可以开始。我们的主人似乎已经进行了充分的学习，对这些理解的比较透彻，所以这天他就叫超级牛力来帮忙准备好这些工具和材料。为了方便用户们安装开发环境，超级牛力已经把创造程序需要的工具都打好了包，包名就叫做 build-essential。所以主人运行了 sudo apt-get install build-essential，超级牛力就去把这些东西从网络上拖回来并且安装好了。

要想写程序，首先得有个文本编辑器。您别听着这个名词觉得很高深，其实您早就认识并且使用过了，像是我们这里的 gedit 小弟，或者查皮那里的记事本，都是文本编辑器。gedit 小弟别看个头不大，论本事可比查皮那个记事本厉害多了。至少人家能够认识一些基本语言格式，什么 C 语言阿，脚本语言阿之类的。用 gedit 打开一个 C 源码文件的话，他会将程序中的一些关键词，常量，变量之类的用不同的颜色显示出来加以区分，这样看起来就比较清楚。

有人可能要问了：我在 Windows 下编过程序，也没用什么文本编辑器啊。只要装好 VC，在里面写就行了，没听说过要用记事本编程的。其实，你使用编辑器了，只不过不是独立的，而是集成在 VC 中的文本编辑器。他们 Windows 的软件就是这样，总是爱搞成个小团体，这 VC 好歹也是有点软公司的招牌软件嘛，对查皮系统中有什么基本的软件应该了如指掌，可明明知道有现成的记事本不用，非要自己带着一个编辑器来，这不是浪费资源么。这就不像我们 Linux，每个软件都顾全整个系统，从大局出发考虑问题。不过话说回来了，那个记事本的能力，也实在是没法写程序用，连个颜色都改不了，真要是写出程序来也得把人看晕了。gedit 小弟比记事本强点，不过其实也有限，写点小程序还行，真要是大程序，那还是有些应付不了，这时候就需要更强大的文本编辑器了。

不过有一点要注意：千万不要问谁是 Linux 下最强大的文本编辑器！

一直以来，在 Linux 这片自由的天空下，有两位公认的顶级的文本编辑器，一位是 vi，一位是 emacs。这两位谁也不服对方，都觉得自己才是空前绝后旷古铄今的全能文本编辑器（说到底还不是文本编辑器而已）。一旦有谁质疑一下这个“最强大文本编辑器”的地位，他们两个都会第一时间跳出来，相互指摘对方的缺点，以确立自己在文本编辑器领域的不败地位。

vi 总是指责 emacs 说：“那么多的快捷键，记忆起来多麻烦。”

这时候一般 emacs 会反驳：“你呢？那么多命令难道容易记？”

“初期需要记住的命令确实多一些，但是总共就那么几个命令，记住之后就可以应用自如了。通过简单命令的组合可以实现各种复杂的操作。哪像你，每种操作都有快捷键要记忆，而且还分那么多模式。每个模式都有特定的快捷键，搞的人晕头转向。”

“你还好意思说我模式多？你自己最大的问题就是作为一个编辑器，还分什么编辑模式和命令模式。搞的新手不知如何是好，连退出都不知道怎么退出。你觉得我模式多？那是我灵活，那是我功能多。你能看邮件么？你能编写网页么？我都能，并且还远远不止这些。”

vi 会冷冷的说：“是啊……所以你才不是最强大的文本编辑器，因为你压根不是文本编辑器，你是个绑定了文本编辑功能的操作系统。”

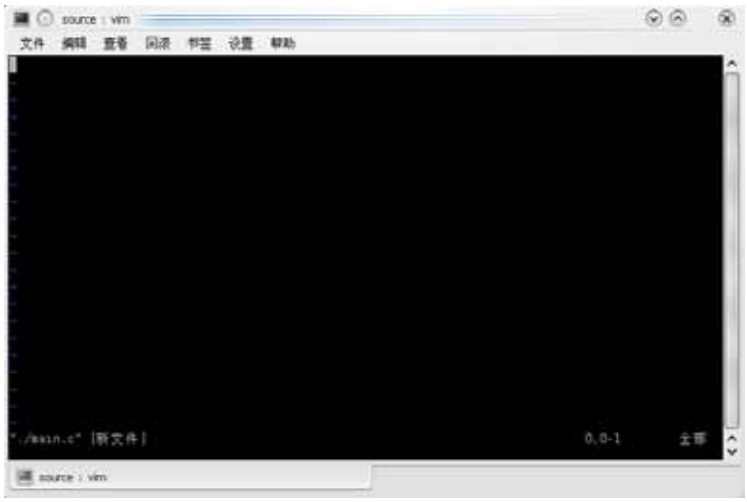
“你连个图形界面都没有你还真好意思说我功能多？”

“那是我简洁，我省资源，我通过串口都能用，你行么？”

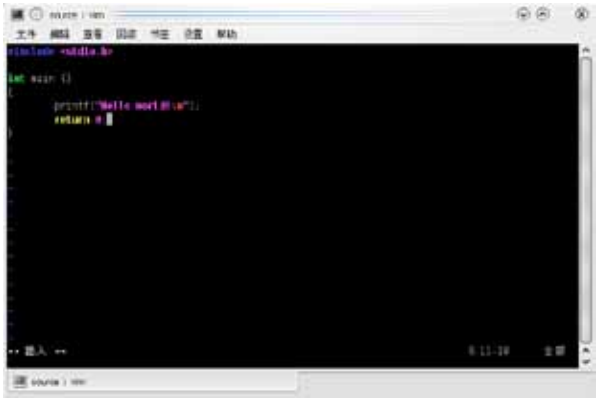
……

哎～总之呢，一定不要让这俩个遇到一起，更不能在有他们两个的时候提到谁是最好的什么什么之类的话题。否则就是：吵不关机死不休！

总的来说，这两个软件是各有千秋，用谁都行。主人选择了 vi，不过，我默认自带的这个 vi 是很古老很难用的，需要安装一个不那么变态的版本才好，他叫做 vim，怎么装就不用我多说了，主人很清楚：sudo apt-get install vim。几分钟之后就装好了，主人首先是创建了一个叫做 test 的目录，然后 cd 进去，运行：vim main.c



进入 vim 的主界面之后，默认是在命令模式，这时候不能够输入任何东西，要按下 insert 键，或者 i 键进入编辑模式。进入之后在下边就可以看见“插入”的字样，这时候就可以输入文字内容了。



“：”表示要输入命令，后面是具体的命令或命令的组合。w 就是保存的意思，主人输入了“:w”后按下回车，这个 main.c 文件就保存好了，

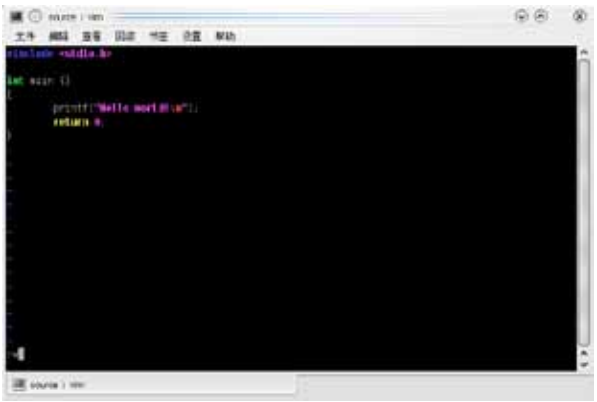
之后主人又输入“:q”，这是要退出的意思，其实也可以直接输入“:wq”，就是保存后退出，这大概就是 vim 所说的命令的灵活组合了吧。

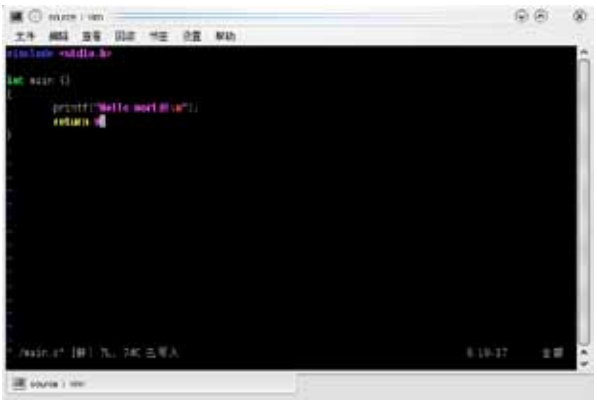
到目前为止，一个简单的 C 程序的源码已经完成了，但是这只是第一步，离真正运行起来还远呢。一个 C 的源程序就好像是盖房子时候的图纸一样，有了图纸就可以了解整个建筑的结构，怎么建造，什么材料之类的。但是光有图纸是没用的，要想把图纸变成房子，需要施工，这时候就需要一个能够把图纸变成房子的施工队。我们这个系统里也有这么一个能够把源码变成程序的团体，他们就是以 gcc 为首的“施工队”。

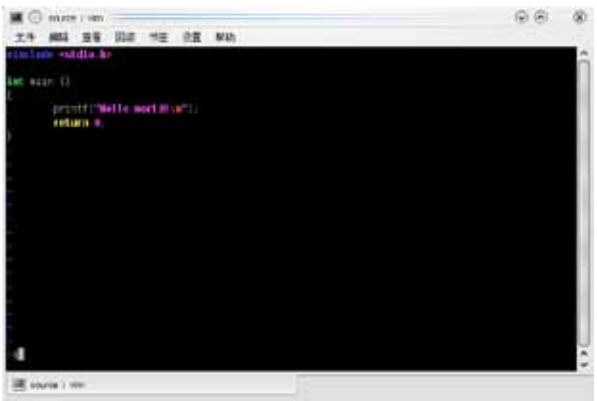
施工队主要成员有 gcc、cpp、as、d 四个人，其中 gcc 是老大，其他几个干什么活都得听他调遣。主人一般也只跟 gcc 打交道，当写好了图纸之后（也就是源代码啦，就比如刚才主人写的 main.c），就

主人进入编辑模式，弹指如飞，瞬间，一个经典的程序写成了，就这样：

主人写了这一小段程序之后，想要保存，于是按了下 esc 键，这是要从编辑模式回到命令模式，因为只有在命令模式下才可以作保存、打开、退出这类的动作。进入命令模式后，主人输入了“:w”，这时候输入的这两个字符可不会在上面出现了，而是出现在左下角。







直接把图纸交给 gcc 去处理就好了，gcc 会去调动其他人进行各种处理。

一般来说，gcc 拿到图纸后，会首先叫来 cpp 进行预处理。预处理主要就是将文件里的宏定义进行展开。什么是宏定义呢？主人一般都比较懒，或者说，他们人类能力有限，不愿意写好多重复的，类似的东西，就把这些都定义成宏。比如，这么写：

```
#define  TOTAL_NUMBER  18353226
```

就是定义总数为一千八百三十五万三千二百二十六，那么以后再要用这个总数的时候，就直接写 TOTAL_NUMBER 就好了，不用写那一大串数字。而且，如果总数变了，只要在最初 #define 的位置修改一次就可以，反正就是为了偷懒。那么 cpp 的任务就是把这类的宏定义都替换回去，把所有的 TOTAL_NUMBER 都替换成 18353226，否则他们老大 gcc 看不懂，老大看不懂，那就没法继续往下干了，因为经过 cpp 预处理之后的文件就要交给 gcc 去编译了。

编译又是怎么个意思呢？最初的图纸，也就是没有经过预处理的源代码，是人写的，一般懂相关语言（比如 C 语言）的人都能看懂。预处理之后的文件，虽然不那么直观了（TOTAL_NUMBER 看着是不是比 18353226 直观？光写个 18353226 还以为是谁的 QQ 号呢），但终究只是做了下替换，还是人类可以看懂的。这样的代码经过 gcc 的编译之后，就不是普通人类可以看懂的源代码了，而是只有终极牛人才能读懂的汇编代码。汇编代码就比较贴近底层的机器码了，里面描述的都是一些基本的操作。打个比方吧，就比如描述切菜的过程，用 c 语言描述出来就像是“将黄瓜切片”，这么一句就搞定了。要是用汇编，那就是：

左手扶住黄瓜，右手拿起刀，移动刀到黄瓜顶部，刀落下，刀抬起。刀向黄瓜后部移动 4 毫米，刀落下，刀抬起。刀再向黄瓜后部移动 4 毫米，刀再落下，刀再抬起。放下刀，走出厨房，走进卧室，找到创可贴，贴在左手食指上……

总之，汇编是一种面向机器的，很复杂的程序设计语言。gcc 的任务就是把 c 语言的源代码转换成贴近机器语言的汇编代码，为下一步 as 的工作做好准备。

as 拿到汇编代码后，对这样的代码再进行处理，得到真正的机器码，这个过程，也叫汇编。进行汇编之前的汇编代码是终极牛人更牛的人才能看懂的，那么汇编之后得到的机器码压根就不是人才能看懂的。汇编程序中至少还有些操作的助记符，比如什么 add 啊，mov 啊之类的。寄存器也是有名字的，比如叫 A，叫 R1 之类的。但是到了机器码，这些都没有了，这些都换成了各种各样的数字，半句人话都没了。还说切黄瓜的事，要是用机器码来描述，那就相当于说：

用 32 号设备扶住 87 号物体，24 号设备拿起 126 号物体，移动 126 号物体到 87 号物体顶部，做 2635 号动作，再做 2636 号动作……

好了，现在终于得到机器码了，机器码按说就是可以执行的代码了，但是，这时候的程序还是不能直接执行的，为什么？因为还有 ld 没有出场呢，他的工作叫：连接。光是一段机器码扔给机器去执行，机器照样摸不着头脑。而且，很多时候，一个程序不是一段机器码，而是由很多段机器码组成的，这些机器码分别存成很多的 .o 文件，这时候就需要 ld 出场了。ld 负责把这些机器码组装起来，并且写明了各段代码的地址，从哪里开始执行之类的。就像我们造个机器人，脑袋啦，胳膊啦，大腿啦之类的都做好了，ld 就是负责组装的。就算只有一段机器码，也就是只有一个 .o 文件，也要由 ld 进行一下处理，闹明白哪是头哪是尾，才能开始运行。

别看说的这么复杂，其实这些过程都会由 gcc 全权负责，主人并不需要操心，主人只需要跟 gcc 交流就好了，交流的方法也简单，就是运行一下 `gcc./main.c` 就可以了。

运行之后，以 gcc 为首的施工队赶紧起床，四个人拿起主人设计的图纸看了看，相互点点头，立即开工，流水线作业：cpp 先把图纸拿过来预处理，之后扔给 gcc 去编译，gcc 再递给 as 进行汇编，最后 as 交给 ld 进行连接，全部完成后，一个崭新的程序出现在了硬盘里。名字叫做 a.out——因为主人没有给他起名，gcc 就默认叫他 a.out。我们内存里的软件们都很好奇的凑过去看着这个刚刚出生的软件，只见他只有不到 10k 大小，看来主人的图纸里基本没有什么内容，也不知道这个家伙会干什么。我们正在胡思乱想呢，主人马上下达了命令 `./a.out` 这就是说要叫醒当前目录下的 a.out，也就是刚刚 gcc 施工队创造出来的这个程序了。

我慢慢的走过去，捅捅还冒着热气的 a.out（刚出锅嘛，可不冒热气，呵呵），温柔的对他说：这位新同学，起床干活啦。只见他一听我叫他，立刻飞身跳进内存，跑进内存后大喊一声：“I am a Rubbish!”然后，跑回去继续睡觉了。 -_-b 我说主人呐，您设计的这程序还真是挺谦虚哈。

后来我们都管这个程序叫做 Rubbish1 号。虽然他能干的事情不多，不过主人还是很满意，毕竟是自己第一次成功的创造了一个程序嘛。很快主人又拿来 Rubbish1 号的图纸改起来。10 分钟后，又把图纸交给施工队，施工队的哥儿几个凑在一起拿过图纸

来看了看，点点头，立马开工，流水线作业，cpp 预处理→ gcc 编译→ as 汇编→ ld 连接，之后，Rubbish2 号诞生！毫无悬念的，主人马上让我叫醒 Rubbish2 号起来干活，于是我走过去叫醒他，只见 Rubbish2 号立刻飞身跳进内存，跑进内存后大喊一声：“I am a Ru~Ru~Ru~Ru~Rubbish~~~~!”然后，跑回去继续睡觉。主人成功的用 for 循环创造了一个结巴。15 分钟后，Rubbish3 号的图纸再次毫无悬念的完成，图纸交给施工队，施工队哥儿几个凑在一起拿图纸看了看，点点头，开工，流水线作业，cpp 预处理→ gcc 编译→ as 汇编→ ld 连接……很快，Rubbish3 号诞生，然后我去叫醒他，然后他立刻飞身跳进内存，对 metacity（Gnome 的窗口管理器）说：“我要一个窗口”。metacity 赶紧给他画好一个，然后他对着窗口喊“I am a Ru~Ru~Ru~Ru~Rubbish~~~~!”，然后毫无悬念的又回去睡觉了。图形界面的结巴……

当当当～～，Rubbish4 号诞生，他这回没让 metacity 画窗口，而是在终端打印出了一句话：“Please Input a Number:”，然后就等着主人输入。主人输入了两个数：5 3，然后 4 号就大声喊：“I~ I~ I~ I~ I~ am a Ru~Ru~Rubbish~~~~!”——程控结巴！

5 号闪亮登场拉～他进来之后，紧闭双目，念动咒语“唵木哒咪咪呀～～分！”然后只见白光一闪，边成了两个！两个 5 号同时喊：

“I am a Ru~Ru~Ru~Ru~Rubbish~~~~!”

“I am a Ru~Ru~Ru~Ru~Rubbish~~~~!”

二重结巴！

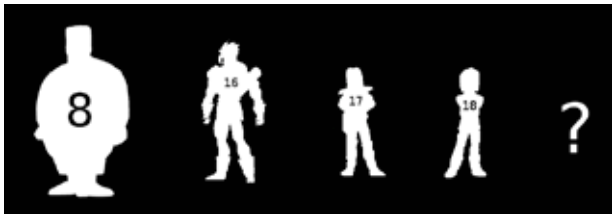
5.2 修理工

“2010 年 9 月 18 日 变天

终于又到周末了。今天在自己的机器上学习编程，逐渐的开始入门了。这一阵子最大的收获是学会了自学。通过网络，论坛，搜索，也能学到不少有用的东西。像学编程吧，开发环境的搭建，循环结构，文字输入，甚至创建进程，都尝试过了。看来编程也不是那么的困难嘛。不过我也知道这只是刚刚入门而已，真的写出个能用的程序和随便写个小程序玩玩还是有很大区别的。

今天我家的楼停电了，不知道哪里的问题，几家邻居慌手乱脚的也整不出个头绪来，最后物业的修理工来了，三下五除二就找到问题所在，没 5 分钟就搞定了，专业就是专业啊。“

今天起床的时间似乎比平时晚了点，全体起床之后，主人按部就班的叫来 OO 老先生记录下一些文字，之后又继续去创造 rubbish 去了。这一段时间里，我们的主人真是笔耕不辍，哦，不对，应该是键盘不辍才对，先后制造了 18 个 rubbish 程序。不过 13 号以下的基本都被主人删除了，只留有一个 8 号。因为 8 号比较憨厚，性格温顺，不爱打架；然后印象比较深的就是 16 号。16 号很爱静，不爱多说话，有点冷漠，但是做起事情来一丝不苟，严格的执行命令；17 号很厉害，但是很自大，高傲，总跟别人发生矛盾。他好像和 18 号还有点什么关系，具体的我就不知道了；最后的 18 号是个美眉，长的很可爱，一头金发，本事也不错，我们大家看他都很顺眼。现在主人正在制造 19 号，不知道为什么，我总觉得即将到来的 19 号将是一个邪恶的坏家伙……



没过多长时间，19 号出炉了。只见他起床之后，跑进内存，刚说了几句我是 rubbish19 号之类的话，就开始乱动别人东西，一会要去狐狸的内存空间里拿数据，一会又要往心有灵犀的地盘里存东西。当然，他的这些企图都没有得逞，要是连这样的小流氓的管不了，我还叫内核么。我们这工作间里面的空间管理是很严格的，谁的空间就是谁用，别人不能乱动。向 19 号这么目无法纪，影响他人工作的软件是不能容忍的。眼看着这邪恶的 19 号，和满工作间无辜受害的软件们，我终于忍无可忍，为了工作间的安宁，为了我稳定内核的荣誉，为了爱与正义，为了部落，为了世界和平，我代表月亮，我，我祭起屠龙刀，看准 19 号，手起刀落，只听的咔嚓一声——整个世界安静了，19 号被我斩为两段，然后我向主人汇报：您的程序出现了段错误。（因为他被砍成两段了，所以错误了）

主人似乎有些不明所以，不知道这个段错误是怎么回事，（因为太血腥了，所以我没直说是因为被我剁成两段。）于是就赶紧叫来狐狸上网上查去。通过搜索知道了，段错误的情况有很多（很多种不老实的程序都会被我砍成两段），但大致上都是由于内存指针使用不当引起的，比如没有给指针赋值就去使用，或者虽然赋值但是访问越界等等。总之就是动了你不该动的内存就会段错误。可是到底这个 19 号是如何动了别人的空间呢？到底他为什么要去访问非法的地址呢？这些情况虽然我们内存里的软件们看得一清二楚，铁证如山，但是主人不知道，他没发钻进内存里来看。那么主人能有什么办法看清楚 19 号的一举一动呢？这时候就需要我们的软件修理工 gdb 闪亮登场～

gdb 是一个字符界面的调试工具，用过 vc 的应该知道在那里面调程序的时候可以进入 debug 模式，单步执行之类的。我们 Linux 中，每个软件秉承着“只做一件事情，但做到最好”的原则，将调试这件事情交给了 gdb 来完成。gcc 编译出来的程序，可以通过 gdb 来运行，运行的时候，就可以设置断点，单步运行，查看变量，查看堆栈等等操作。有了 gdb，主人就可以监视程序在内存里面的一举一动了。不过 gdb 并不是像狗仔队那样想监视谁监视谁，像狐狸啦，gedit 这样的成品程序是不能被监视的，要想让某个程序被 gdb 监视，必须在制造他的时候——也就是编译的时候，留出给 gdb 控制的接口来，gdb 才能监视那个程序的一举一动。看过黑客帝国么？我们机器里的普通的程序就像是里面的正常的自然人，而可以被 gdb 调试的程序就像 Matrix 世界中的人一样，脑袋后面有个接口，可以接进去控制。那么怎么给程序装这么个接口呢？很简单，就是在编译的时候加上参数 -g。

```
Gcc  -g  ./main.c  -o  rubbish9_debug
```

主人运行了这么一句，创造出了脑袋后面有接口的 rubbish 19 号。之后就叫来 gdb 去运行他，就这样 `gdb  ./rubbish9_debug`

```
GNU gdb (GDB) 7.0-ubuntu
Copyright (C) 2009 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i486-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/saub/test/rubbish19_debug...done.
(gdb)
```

gdb 接到命令，赶快掏出各种仪器和工具，并把 19 号拖进内存里，然后从一个大机器上抽出一个长长的电缆，插进 19 号脑袋后面的接口里，一切准被好之后，向主人报告：一切准备就绪，可以开始了。

主人好像是之前看书学习过了，虽然就我所知这是他第一次使用 gdb，但是似乎还挺熟练，看到 gdb 的提示后，输入了 r，并且回车，意思就是：开始运行。（run 的首字母）于是，gdb 按下一个电钮，19 号的身体跟着腾的一下站了起来，启动了。插着电缆的 19 号像他上次启动一样，还是要去骚扰狐狸妹妹，访问人家的内存空间，于是我也只好举起屠龙刀，再次将其一刀两断。gdb 赶快向主人报告：19 号同学在按照您的图纸进行 xxxx 个动作时候的，由于侵占他人内存空间，触犯了内存治安法第 287 条，因此被处以断刑。然后还指出了 19 号的这种

```
Starting program: /home/saub/test/rubbish19_debug

Program received signal SIGSEGV, Segmentation fault.
0x080483f4 in main () at ./main.c:6
18*msg='R';
(gdb)
```

行为在主人的图纸中所在的位置，也就是代码行数了。

主人一看，18 行就出问题了，很打击自尊阿，可是这 18 行也看不出什么不对的来。应该是这个指针有什么问题，还是从头看看程序吧。于是主人输入了 list 命令，gdb 赶紧把 19 号的整个图纸——也就是全部程序的源代码打印了出来。程序一打印出来，我们就都看明白了，这个 msg 指针压根就没有初始化嘛，只是做了声明而已，也没给申请内存空间，就这么用，那不出问题才怪。

虽然我们都看出来了，但是主人似乎还是没明白过来，于是还不断的用 b <linenum>，这样的来设置断点（break 的首字母），然后用 n 命令来一步一步的运行（next 的首字母）。直到他用 p 命令（print 的首字母）来打印 msg 的值的时候才终于意识到：哦，好像没给分配空间哈，msg 的值一直是 NULL 呢。

故事外的事——内存管理

这一回跟您介绍了主人创造的 rubbish 19 号程序，这家伙由于不遵守《Linux 系统内存管理条例》被我干掉了。那么我们 Linux 的内存管理是怎么一个原则呢？都有什么规矩呢？下面我就跟您说说。

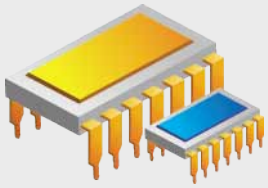
一个程序工作的时候需要用到的内存分为几个部分：代码段，数据段，BSS 段和堆栈段。其他的基本上程序一启动就确定好了的，没什么可说，也没太多要管理的，唯独这个堆栈里面的堆空间，是需要管理的。

堆空间是指程序在起床运行后向我申请来的空间，也是一个程序占用的最多的空间。一个程序如果要想使用工作间里的空间，他要向我提出要求，说我需要多大多大的一块内存空间——这个过程叫做申请。然后我根据工作间里的情况来分配，告诉他，哪块哪块归你，然后这个程序就去用去了。这时候那块地方就单独给这个程序使用，不许别的程序访问了。如果别的程序胆敢来访问这块空间，就像你去人家偷东西一样，必须依法剁成两段。（偷东西没这么大罪过吧……）

那么这块内存空间分给这个程序使用之后就永远给他了么？想的美！你买房还只有 70 年产权呢，这么珍贵的内存空间怎么可能永久分给一个程序。申请倒内存空间的这个程序在做完了相关的事情，不再需要这块空间的时候，他应该跟我报道，说空间我用完了，这块地方可以再给别的程序用了——这个过程叫做释放。

一个有知识有道德有理想的程序，在他回硬盘睡觉以前应该释放掉所有他申请过的空间的。如果遇到哪个不良程序，无德青年申请了很多空间不释放，我也没有办法，因为这不像越界访问那样，访问了别人的地址那就是访问了，人赃俱获，无法抵赖。人家不释放也许是因为他一会还需要用这块空间呢，我没法证明他申请的空间用不着了。我唯一可以做的就是在他回硬盘睡觉的时候，检查所有他申请过的空间，如果有没释放的，就强制释放掉——你都睡觉去了，你申请的空间肯定用不着了嘛。但是如果这个程序是个长时间运行的程序，并且还不断的申请新的空间而不是放，那就麻烦了。内存空间会被他一点一点的消耗光，这就是最烦人的内存泄漏，在我们软件界，内存泄漏是和瓦斯泄漏同样严重的事故。《Linux 系统内存管理条例》第三条明确写着——禁止申请不释放！就在第四条禁止抽烟的上面。（当然不能抽烟，内存都冒烟了机器还能用么？）

其实不光是 Linux，查皮那里的程序一样需要遵守类似的原则，估计他们那里大概也有个什么《Windows 系统内存管理办法》之类的文件吧，反正大家的原理是一样的。



不过对于工作间的使用，查皮和我还是有点不同的。查皮总是喜欢尽量留出空间来，好给新起床的程序用。可是我总觉得，作为一个系统，我怎么能知道主人还会有什么程序要运行呢？要是没有程序要来了，工作间里还空那么大地方，不让正在工作的程序用，那不是浪费么？

我还是习惯尽可能的把东西都搬进工作间里。除了程序们申请多少内存就尽可能给多少之外，剩下的部分，我就把一些可能会用到的库啊，命令啊啥的统统都搬进来，能占多少占多少。那有人问，要是你把这里边都沾满了，待会有程序要进来咋办？很简单啊，我再搬出去呗！程序要进来，也不是一下子都进来，他也得把他的东西一点点搬进来，他往内存里搬的时候，我就往外搬，不耽误。

所以，当有程序要启动，跟我说：我要 10 平米的地方放东西。我就先答应他说，好，放吧，有地。然后在他往里搬的时候我再给他腾地方。也可能他要 10 平，但是先只用了 2 平，那我就先腾出 2 平来，等他再要我再腾。他们管我这个方法叫 Copy-on-write。

查皮就不同了，可能是因为他比较胖的缘故吧，他比较懒，不愿意搬来搬去这么折腾。基本上他只是在必须用啥东西的时候才把那东西搬到内存里，让内存留出尽可能多的空间。这样，当有程序管他申请内存的时候，他就可以用手一指：那块地，归你。然后就不用管了。实在内存不够用的时候就找个比较闲的程序，命令他：你，去硬盘里先忍会。

5.3 包工头

“2010 年 10 月 10 日 起风

今天好孤独，节后的只有一天的周末还要独自来加班。领导总是不顾员工的死活，就像剥削劳动人民的包工头一样。他让我无论如何要解决全公司计算机的杀毒问题，不管怎么解决，反正必须解决。这叫什么事嘛，也不调查一下实际情况，全公司那么多台电脑，我怎么给他们挨个杀毒？保证服务器不受病毒干扰那是我的责任，没的说，可我怎么能保证每一个员工的电脑不中毒呢？谁知道他们每天都拿电脑装些什么乱七八糟的东西！真是气死我了。”

最近硬盘里的 rubbish 越来越多，已经排到 31 号了。虽然这些程序都不大，31 个加在一起也就几 M 的大小，可关键是这帮不着调的程序要是运行起来就乱了套了，指不定哪个就给你捅娄子玩。你像 19 号这个段错误倒是还好说，我能看得见，并且能第一时间让主人知道，主人也就想办法修改了。可是比如 23 号，有内存泄露，这个我没发第一时间通知主人，也加上我们这 4G 内存足够大，泄露了主人也感觉不出来，所以一直也没有修正这个错误。还有那 24 号，总是乱动别人的文件，那次就把狐狸妹妹的文件给弄乱了。人家狐狸妹妹工作的时候有许多东西要记录的，比如网页用什么字体显示啦，主人喜欢去那些网站啦之类的东西，狐狸妹妹都会写成配置文件存放在自己的那个目录里。结果那天那个 rubbish24 号一进工作间就上窜下跳，整的大家都不得安生，来来去去的要创建文件、修改文件、删除文件。你说你自

己折腾着玩谁也不管你，可是等他折腾完了会去睡觉了才发现，狐狸的配置文件让她给弄没了。闹得狐狸跟失忆了似的，主人再打开她的时候，模样也变了——因为不记得主人喜欢什么样子的了。主人说：去我最常去的那个网站。她眨眨眼问：哪啊？把主人气得鄙视了狐狸半天。我们真替狐狸冤枉阿。还有那个霸占着 CPU 不放的 27 号；在屏幕上乱写乱画的 17 号等等，罄竹难书阿～。结果今天 vim 传来一个更悲惨的消息：主人觉得这堆 rubbish 程序都没什么实用性，要把 31 个 rubbish 合并起来，创造出一个能陪主人消磨时光的，有智力的 rubbish 合集！！我们一听头就大了，这主人是不是变形金刚看多了阿，31 个不着调的疯子程序合体能合出什么好玩意来？赶紧问 vim：消息可靠么？vim 说：可靠，主人打开了最初的那个 main.c 文件，正在把其他的 rubbish 的代码往里面拷贝，我们一听，完了……

只见主人首先拿来 rubbish1 号和 2 号的图纸，由于 1 号实在没什么本事，而且跟 2 号的动作差不多，于是就把这张图纸改成了整个合体后的 rubbish 的总体结构，（咱就把合体后的 rubbish 叫做 Rubbish 金刚吧）然后将 2 号的程序作为 rubbish 金刚的一部分加入到 rubbish 金刚的代码中。然后主人尝试编译 gcc main.c -o all_in_one，然后运行 all_in_one，效果不错。

之后主人继续让 rubbish 合体，要将 3 号的代码加进来。可是主人自己一想，这么多代码，要是都写在这一个 Main.c 里面多乱阿，对了，我把他们按照功能分别写成独立的 .c 文件吧。于是主人把代码结构整理了一下，设计了这么几个文件 main.c，board.c，board.h，ai.c，ai.h。

看这名字还真猜不出来主人要创造的这位 rubbish 金刚是干啥的。但估计不是什么好鸟，我们还是小心为妙。这样把结构调整了之后，主人再编译就要：gcc main.c board.c ai.c -o all_in_one，然后运行 all_in_one，恩，程序正常——因为 board.c 和 ai.c 里面还没什么东西呢，main.c 里面也不过是点打印语句当然正常。

再继续，主人继续把之前的一些 rubbish 的图纸往 board.c 和 ai.c 里面复制，复制了一点之后进行适当的修改，然后就编译试试。每次主人一编译，施工队那哥几个都得跑过来，把三张图纸都打开，拼在一起，然后开始从头施工。有时候主人只是在 ai.c 里面修改了很小的一点东西，编译的时候施工队的同志们也要从到尾的重新编译每一个部分。这就好像盖个楼，完工之后开发商说：一楼这个大门的门把手图纸上画错了，应该用圆的，怎么画成方的了？改了吧。施工方一听，赶紧下令：图纸画错啦！把楼炸了重新盖！虽然这样对于拉动 GDP 发展有很好的作用，但毕竟属于精神不正常的范畴。那么我们的 gcc 施工队为什么做这种很抽风的事呢？这不怪施工队，这是因为主人这么写的命令：

```
gcc main.c board.c ai.c -o all_in_one
```

就是要这么干，施工队也只是服从命令而已。所幸我们的主人觉悟还是比较高，神经还

算基本正常，很快就意识到了这么做可能有点浪费劳动力。三个文件里，改了哪个，就只编译哪个可以么？当然可以。这回主人这样编译：

```
gcc -c main.c board.c ai.c
```

这是告诉施工队，只把这些图纸分别编译成模块，并不组装起来。施工队按照指令施工完毕之后，目录里没有什么 all_in_one 程序，而是出现了 main.o board.o ai.o 三个文件。之后再 gcc main.o board.o ai.o -o all_in_one，才生成最终的程序。这样如果主人修改了 ai.c，只要运行 gcc -c ai.c，重新生成 ai.o 就可以了，board.o 和 main.o 就不用动了。这样施工队倒是省了不少事，可是主人又不平衡了：以前我只用一条命令 gcc main.c board.c ai.c -o all_in_one 就可以了，这下我每次修改之后都得 gcc -c xxx.c，然后 gcc xxx.o xxx.o xxx.o -o yyyy。这不是反倒麻烦了么？

哎，看来主人还真不是个搞编程的，要不怎么连这么点问题都不明白呢？这样编译虽然看上去多打了一次命令，但是节约的编译时间阿。现在主人的程序小，体会不到，要是编译个内核呢？明明只改了一点点，却要编译整个内核，浪费将近半个小时？再说，其实这些命令完全可以不让主人每次都敲的。那 gcc 施工队他们毕竟只是个施工队，你要是盖个小厨房，垒个猪圈啥的，这样的小东西直接找他们就没问题。直接一编译：gcc 砖头 -o 猪圈，这就出来了。可现在你要搞一个无敌全能不着调的 rubbish 合体，这可就不是猪圈了，这就是个 CBD 商圈，里边什么银行啊，商场啊，写字楼啊，炸油条的，卖臭豆腐，修理自行车的等等，一应俱全。这么大的一个工程，你光叫个施工队来那肯定搞不定阿。这得有人进行合理的统筹规划，设计施工方案，然后再让施工队去具体施工。而这个规划的人谁呢？按照主人现在的做法，这个规划人就是主人自己，但他自己又没这本事，怎么办呢？这时候他就需要专业的规划人——make。

make 也是一个程序，像上边说的一样，他就是负责控制整个施工过程的（也就是编译过程啊）。对于比较小的程序，就一两个 .c 文件，那根本用不着 make 出马，直接 gcc 施工队去编译就行了，因为源文件的结构关系不是很复杂。可是要稍大一点的程序，像狐狸妹妹啊，心有灵犀啊，星爷啊，基本上所有常用的软件吧，都足够复杂到需要 make 来对编译过程进行管理。当软件大了，编译的时候就不能像主人刚才似的简单的把一大堆 .c 的源文件统统一次性编译成一个二进制文件那么简单了。那样做的话费时费力，主人现在虽然还没有太体会到，但是以后要是程序再大点，他要想修改一下，可就费劲了。

比如说吧，有一个软件，源码由 20 个 .c 文件组成，分别是 1.c、2.c、3.c……20.c。这 20 个文件一股脑都交由 gcc 包工队，他们就会把这些文件都打开来，拼在一起，一次性的编译成一个叫做 big 的二进制文件。这时候发现了一些问题，需要修改 3.c 文件，修改之后得重新编译啊，那么 gcc 包工队又得把这 20 个文件全都打开，拼在一起，再从头到尾编译一次。而其实只有 3.c 文件修改了，完全不必这么兴师动众。那应该怎么做呢？一般的都是把这 20 个文件分别编译成 .o 文件，比如编译成 1.o,2.o,3.o……20.o，这样 20 个 .o 文件，然后再由 ld 把这些 .o 文件拼在一起，成为一个叫做 big 的二进制可执行文件。那么当要修改 3.c 的时候，只需要让 gcc 包工队重新将 3.c 编译为 3.o，再让 ld 重新连接一遍就好了，省去了很多时间。

而这个过程，如果让主人自己管理的话，会很麻烦，毕竟他们人类的大脑也不是那么靠谱的，搞着搞着就乱了。于是，make 义无反顾的挑起了这个重要的担子。当然 make 也不能靠凭空的想象就来指导包工队干活，什么事情总得有个规划不是。make 也需要一份施工的规划书，这分规划书就是 Makefile。

Makefile，顾名思义，就是 make 用的 file。这就相当于一份施工的规划，上面写着整个工程分为几个模块，先用哪几个文件编译成一个什么什么 .o，再用哪几个文件编译出一个 .o，再怎么怎么一连接，最后得到编译好的二进制程序。make 就根据这份文件来指导 gcc 他们进行施工。当有某个 .c 文件被修改之后，make 能够根据文件的修改时间智能的判断出哪些模块需要重新编译，重新连接，然后就去让 gcc 重新编译那些改过的文件，最终生成新的二进制程序。有了 make 和 Makefile，就省去了主人敲一大堆编译命令的烦恼，只要敲一个 make，其他的，就交给 make 去做吧，他办事，你放心。

vim 很快又来报告：主人新建了一个叫做 Makefile 的文件。

恩～看来主人终于想起 make 这个软件了。我刚要欣慰一下，忽然一想，不对！主人越是用了合适的软件，那个 rubbish 金刚就越早的来工作间捣乱阿！哎呀，怎么办啊？怎么办啊？可是咱们也不能拦着主人干活不是？只见主人用 vim 打开并创建了 Makefile 文件，然后在里面写道：

```
all:main.o  board.o  ai.o
gcc  main.o  board.o  ai.o  -o  all_in_one

main.o:  main.c
gcc  -c  main.c

board.o:  board.c  board.h
gcc  -c  board.c

ai.o:  ai.c  ai.h
gcc  -c  ai.c

clean:
rm  ./*.o
rm  all_in_one
```

这就是专门给 make 看的 Makefile 的内容，也就是相当于施工流程说明。这里面写了，最终的目标，是编译出一个叫做 all_in_one 的程序。要创造这个程序，就需要 main.o,board.o,ai.o 这三部分内容。有了这三个文件后如何创造呢？就是运行：gcc main.o board.o ai.o -o all_in_one。这也就是最上面，all 那一段写的意思。下面又分别写了如何建造出 main.o,board.o, 和 ai.o 这三个文件。比如 main.o 这段，意思就是，要想得到 main.o，那么需要 main.c 这个文件，有了这个文件如何得到 main.o 呢，就是运行 gcc -c main.c 这个命令，下面的 board.o,ai.o 也是类似。简单的说，这里的格式就是：

目标：原料

加工方法

但要注意，加工方法那一行必须以 tab 符号开头，而不能是空格（虽然都是看不见的一片白，但是决不能搞错），不知道 make 这家伙为什么有这样的怪癖。

有了 Makefile 之后，主人修改程序之后，编译就变得非常方便了。比如主人改了 ai.c 文件，存好之后，直接运行 make，就可以了。运行之后 make 会跑进工作间，拿起当前目录下的 Makefile 来看这个项目各个文件之间的关系。之后再去检查 Makefile 里面提到的这些文件的最后修改时间，一看就能发现，那个最终的产品，all_in_one 程序的最后修改时间是 3 点 15 分。ai.c 的时间是 3 点 20 分，其他的文件最后修改都是在 3 点 5 分左右，反正在 3 点 15 分之前，于是 make 马上通过严密的推理和判断，知道了——主人在 3 点 5 分左右写完了上一版代码，然后在 3 点 15 分编译出了最终的程序，之后在 3 点 20 分又修改了 ai.c 文件，先在想要重新编译一下。那么本着节约用电，不搞重复建设的原则，我应该只编译 ai.c，把他编译成 ai.o，之后在把新的 ai.o 和刚才已经有了的 main.o 和 board.o 组合成一个新的 all_in_one 程序。于是，make 就去叫来 gcc 施工队，下达命令：gcc -c ai.c 然后 gcc main.o board.o ai.o -o all_in_one。

当然，咱们是说的复杂，其实 make 想这些事情只是电光石火的一瞬间而已，一个 make 命令敲下去，马上就看到施工队们出来干活了，一个崭新的 rubbish 金刚就诞生了。

诞生之后，主人赶紧运行一下 ./all_in_one，果然运行起来了，看来 Makefile 写的很成功。不过主人想了想，看别人的程序，好像 make 完了还得 make install 一下，要不不专业。make 之后程序就编译完成了，这个 make install 又是干什么的呢？赶紧叫过来狐狸妹妹就开始查。我在内存里看得这叫一个着急，你说你找个 makefile 看看不就得了么。再说，make install，这多好理解的名字阿，就是安装呗。我们 Linux 的软件都是按照文件的类别安装进系统的各个目录中的。二进制的文件放在 /bin/usr/bin 这样的目录，库文件放在 /lib/usr/lib 等等。你这个程序刚刚 make 完，只在当前这个目录下生成了二进制文件和各种其他库文件和存档文件。（当然了，我说的是一般的软件，主人这个弱智软件没那么复杂。只有一个二进制文件。）你总不能每次的 cd 进这个目录里然后运行 ./xxxxx 这么用吧，很麻烦的。所以要安装，把二进制文件放进 /usr/bin 目录，库文件放在 /usr/lib，其他相关文件放在 /usr/share 之类的目录，这样你无论在哪，都可以调用这个程序了。那这个 make install 的过程，其实就是把编译好的文件复制到相应的目录去这么个动作，由于这是要往系统目录复制东西，所以需要在前面加上 sudo。还有的文件有 make uninstal，也是一样的道理，其实就是把相应的文件给删除。

主人查了半天终于明白了 make install 的道理，于是又加工了一下他的 makefile，加上了这么一段：

```
install:
cp  ./all_in_one  /usr/bin/
uninstall:
rm  /usr/bin/all_in_one
```

之后，主人当然要试验一下，于是先运行 make clean。make 接到命令，赶紧打开 makefile，看到 clean 一段写的内容，于是照着执行，把当前目录下所有的 .o 文件和 all_in_one 这个祸害给删除了。清除了之前编译的内容之后，主人再运行 make，这回 make 接到命令，一看后面没有明确的参数，就按照自己的习惯在 makefile 里面找 all: 一段。找到之后发现要想完成这段，需要 main.o，board.o 和 ai.o 于是再一次找 main.o:board.o: 以及 ai.o: 三段内容，并依次按照上面些的命令运行，最终得到 all_in_one。（顺便说一句，如果 make 找不到 all: 一段，就会执行 makefile 里写在最前面的一段。）然后主人在运行 make install，make 一看 install: 段，简单，把文件拷贝过去了。主人特意跑到别的目录，运行 all_in_one，果然可以执行，看来是安装成功了。不过主人也知道这么个程序放在系统里不是很合适，于是又回来执行了 make uninstall。

5.4 分析师

“2010 年 11 月 15 日 晴

昨天 mm 来我家，我给她看了我编的 rubbish 合集程序。mm 和那个程序玩了几把‘九宫格’的游戏，竟然还输了一回，说明我写的算法还是不错的，呵呵。看来编程也不是那么难，只要找到合适的切入点，潜心学习，没什么学不会的。编游戏看来是个不错的开始，虽然这个程序界面相当简陋，而且肯定有很多的潜在问题，不过对我自己来说还是个很好的开始。这个 Linux 系统学习编程好像不是很方便，不过它倒是让我能够比别人更了解程序编译的整个过程。虽然我写的程序肯定是很幼稚的，不过今天忽然有个想法——把它发布出去！我的程序也要开源，哈哈。这样让更多的同学们分享我的成果，并且告诉他们，新手也可以写出有用的东西来。”

哎～主人最近是越来越不着调了，他竟然想把那个超级无敌 rubbish 金刚发布到网上去！祸害我们一个系统还不够，还要祸害多少青春懵懂的 Ubuntu 啊。

其实那家伙也干不了啥正经事，主要就是能陪主人玩游戏，所以才这么受宠。说来也怪了，我们软件库里面那么多游戏主人都没兴趣，不知道为什么就对这么个简单的九个格子的游戏情有独钟。不过也难说，毕竟是他自己编的嘛，谁的孩子谁不爱呢。可是您自己喜欢那就偷偷摸摸自己玩就行了，干嘛非要发布到网上让他去祸害别的电脑呢？

这家伙经常的申请了内存不释放，有时候还假死，这要是不明所以的人用了这个程序没准还抱怨我们 Linux 系统不稳定呢。当然了，发牢骚归发牢骚，主人的命令我们还是得执行，这会狐狸妹妹就正忙着把主人的 rubbish 金刚传到网上去呢。

过了一会，狐狸妹妹忍着笑就过来了：“你知道主人怎么发布他那个程序嘛？主人直接把编译出来的 all_in_one 文件，贴到了论坛里面。哈哈，他以为这样直接就能运行呢。笑死了。”恩，看来主人还有很长的路要走阿。这个二进制文件看上去就是单一的文件，但其实他运行起来是需要很多库文件来协助的，不是拿到哪都能运行的。在他们 Windows 界其实也是这样，98 的程序直接拿到 xp 下不一定能运行，xp 的程序也有很多装不到 win7 上。但是由于他们的版本比较少，而且系统的各种库和接口啥的都比较统一，所以很多人觉得程序就是一个 exe 拷贝到哪里都可以运行。主人这个程序，如果拷贝到一个和我们相同的系统上，那肯定是可以运行（也肯定可以造成内存泄露和假死，哼哼）。可是不一定别人的系统就跟你的系统一模一样啊是不？

尤其我们 Linux，发行版是五花八门，就算相同的发行版，版本也不一定一样。如果系统里没有这个程序所依赖的那些库的话，这个程序肯定是运行不起来的。要想知道一个程序依赖于哪些库可以用 ldd 命令来查看。我趁主人不注意的时候叫来 ldd，运行了一下 ldd./all_in_one，看看这个软件都依赖什么。Ldd 向我汇报如下：

```
$ ldd ./all_in_one
linux-vdso.so.1 => (0x00007fff31dff000)
libc.so.6 => /lib/libc.so.6 (0x00007ff61215c000)
/lib64/ld-linux-x86-64.so.2 (0x00007ff6124cc000)
```

看来依赖的还不是很多，算是最基本的了，可也照样不能随便放到别的系统上运行啊。比如这个 ld-linux-x86-64.so.2, 明显只有 64 位系统才可以。libc.so.6 这个虽然是个系统就有，但是版本也得合适，不合适也不行。这也是为什么 Linux 上发布的软件好多都是源码包的原因，因为系统环境实在是各式各样，还是把源码放到目标系统上编译一遍来的方便。然而我这些话也就跟您说说，我主人听不见我说话的，所以他还是把那个 rubbish 金刚直接贴到网上了。

几天之后，主人就收到了应有的回应：“这个程序在我这运行不了啊？”“楼主再好好好看看吧。”“我是 32 位机啊，运行不了，楼主提供个 32 位的版本吧。”…… 主人终于意识到这二进制文件不是放到哪都能执行的，可要是他们每个人编译一个针对他们系统的版本也是不大可能，好吧，那就发布源码！可是问题又来了，源码发布的软件包应该是什么样子呢？为了不再次体现自己的无知，主人从网上随便下载了个软件的源码包，打算来学习一下。主人下的这个软件是 Lynx，这是一个字符界面的浏览器，倒不是主人想要用他，主要是这个体积不大，拿来体验体验编译安装的过程而已。

这个软件包下载下来叫做 lynx2.8.7.tar.gz。这种包是下载软件最经常找到的了。另外还有就是 tar.bz2 格式，跟 tar.gz 的没什么区别，只是压缩格式不同而已，就像一个是 rar，一个是 zip。我经常听到很多其他的笨兔兔抱怨他们的主人围着个 tar.gz 包不知道该怎么办，自己急的直打英文字也没办法。还好我的主人了解的多一点，知道这样的包是怎么回事。其实事情是

这样的：话说有个软件叫 tar，基本上每个 linux 都会带着这么个软件，我这里也是。这个软件是干什么的呢？是个打包裹的，不过他可不是邮递公司的那种，不会把打好的包扔来扔去。他的能力有点像查皮那里的 winzip，他能把很多文件和目录收拾在一起，打成一个包裹，也就是生成一个 tar 包文件。可是跟 zip 不一样的是，tar 只管打包，不管压缩。原来那些零碎的小文件有多大，打成 tar 包之后还是多大，只是变成一个整个的文件了而已。有人说，那我想压缩怎么办？别急，我这还有另一个软件，叫 gzip。这个软件就是专门负责压缩的，但是他只能压缩一个文件，不能像 winzip 那样能压缩一个目录里的好多文件。这样，tar 和 gzip 就成黄金搭档了（有脑白金么？），要想实现 winzip 那样的功能，就得 tar 和 gzip 联手协作。

比如有个目录叫 aaaa，里面有好几十个文件，总共有 10M。想要压成 zip 那样的压缩包，那就先让 tar 出手，把 aaaa 目录打成一个包裹文件——因为 gzip 只能压缩一个文件嘛。这样 tar 就把这个目录打成了 aaaa.tar 文件，这个文件还是 10M 大。然后由 gzip 出场，把这个文件压缩，压缩完了得标明一下啊，所以就又把文件名改了，叫做 aaaa.tar.gz，表示这个文件经过了 gzip 压缩，这时候这个文件就小了，可能 5M，可能 7M 的就没准了。有时候还有叫 xxx.tgz 的包，也是一个意思，只是把 .tar.gz 扩展名合并了而已。这下就明白了吧，这个 tar.gz 包其实就相当于 rar 或者 zip 的压缩包。那下载来的 tar.gz 包的软件怎么装呢？那当 是先把包解开再看了，得先解开压缩包看看里面是什么内容才能知道怎么装啊，就像我问你 RAR 包怎么装，你能知道么？

我们现在知道 tar 包就是个压缩包，就是个大包裹，里面有什么东西不一定。那一般拿到一个 tar 包的软件应该怎么办呢？

你收到一个包裹后怎么办？当然是先打开啦！先找剪子啦，小刀啦之类的工具把包裹拆开，然后看看里面有什么东西，根据里面东西的不同来决定怎么处理。里面要是家里寄来的松子核桃什么的，就赶快吃了；要是比较难吃的松子核桃什么的，就跟同事分着吃了；要是部手机，就赶快拿出来试试；要是下面还有把手枪，就赶紧拿刚才那手机报警。这些大概不用我说，智力正常的人都应该知道怎么做，其实 tar 包也是如此。拿到一个 tar 包之后，先用你的工具把 tar 包拆开。工具是啥？有道是解铃还须系铃人，tar 打的包，当然还用 tar 来解了。

当然，你也可以用那个叫做文档管理器的家伙，他的中文名字叫归档管理器（叫 gui ～ dang ～ guan~li~qi~~？那是小沈阳！），他的英文名字叫 file-roller。不过其实他只是个负责用图形界面和主人交流的家伙，真正干活的还得是 tar。tar 包解开后，一般会得到一个目录，里面有很多的文件。然后干什么呢？有的同学记起来了，看看里面的东西啊。

一般包里面应该有个 README 文件，里面写着这个软件是干什么用的、怎么安装、怎么用、作者是谁、干什么的、爱吃什么、身高多少，腰围裤长……等等信息吧。也可能安装的方法写在一个叫做 INSALL 的文件里。总之，应该有相应的文档文件来告诉你这个软件怎么装。不过也有时候软件的作者不厚道，或者忘性大，没有写 README 或者 INSTALL 文件，或者文件有，但是没说清楚到底怎么装，那怎么办呢？用自己的头脑判断一下吧。

主人把下载来的软件包移动到了他的家目录下，也就是～目录，然后运行：

```
tar  -xzvf    ./lynx2.8.7.tar.gz
```

把这个压缩包解了出来。解压之后是一个目录，叫做 lynx2-8-7。主人运行 cd lynx2-8-7 进入到这个目录里面，用 ls 看了一下，发现有很多文件，其中有个叫做 INSTALLATION 的，里面大约应该是写着安装方法吧。不过主人根本没打开这个文件，而是看到了 configure 脚本，于是用自己的头脑判断，直接运行了：

```
./configure
```

这个 configure 是干什么的呢？

我们知道 gcc 施工队听 make 包工头的指挥，make 包工头根据 Makefile 安排工作。这样，如果想把一堆源码编译成二进制的程序，只要执行一下 make。执行之后 make 会在当前目录下寻找 Makefile，然后按照上面写的方案，指挥施工队：在这盖个大裤衩、在那盖个水煮蛋、再跟中间垒个鸟窝……然后施工队按照命令一点点施工，直到最终完成任务。然而事情有时候并不是那么简单，没准包工头 make 下达大裤衩命令之后，施工队回来报告：这地方挨着鞭炮厂阿，盖大裤衩还不烧着了？包工头说：那先盖水煮蛋吧。施工队又报告：这地方长年干旱，地下水位也底，这点水连泡面都不够，别说煮蛋了。包工头只好说：那就先盖那鸟窝，总行了吧？施工队再说：鸟窝倒是能盖，就是这地方不通天然气，点不着窝里那火炬阿……

遇到这些问题，都是由于开工之前没有对这里的环境，现有的材料进行合理分析导致的，那么我们的这个 configure，就是这样一名分析师。

configure 跟 make 不一样，他并不是常驻在我这里的软件，而是每个源码发行的软件自带的一个脚本。简单点说，幸福的 make 都是一样的，不幸的 configure 各有各的不幸。有了 configure 之后，编译软件的步骤就多了一步——./configure 让这个分析师首先开始工作，他会检查当地的情况，有什么材料啊，什么库啊，什么编译器啊之类的，都检查一遍，然后因地制宜的设计一份 Makefile。如果有足够的水，才允许煮蛋，有远离火种的安全空间才能晾裤衩，如果条件不满足，configure 就会报告错误，告诉主人这里缺少什么，等主人想办法弄齐了再来编译。如果条件满足可以施工，configure 就会出一份 Makefile，注意，一般 configure 调查前，目录下是没有 Makefile 的（当然，没有 configure 的情况另说）。

主人运行 ./configure 之后，configure 对我们系统进行了检查，发现可以施工，于是就生成了 Makefile 文件，主人接着运行 Make，开始编译，由于软件很小，马上就编译完了，最终主人运行 make install, 把这个软件安装在了我们系统里。

有了这么一番感受之后，主人知道了一个源码发布的软件包的大概样子。于是照着人家这个软件包，对比一下自己的这个软件。makefile 是现成的，只要再增加一个 configure 脚本，检查一下系统中有没有 gcc 之类的编译工具，以及库文件啥的版本就好了。要写这么个脚本，那就得学习 shell 脚本编程了，主人之前可没学过，这会现学有些来不及，那怎么办呢？对，照猫画虎，看看那个 lynx 的 configure 怎么写的，跟着学就好了。主人想的挺好，叫来 gedit 打开 configure 一看就傻了！

——洋洋洒洒连注释带语句一共三万六千三百七十九行！这还不得看到猴年马月啊。

5.5 规划局

“2010 年 12 月 20 日 回冷

总算是把代码发布到网上了。很多热心的网友提出了不错的建议和意见。才发现我写出来了的程序还真的很白痴。经过了这么一个过程，确实在编程方面长进了不少，了解了很多以前不了解的事情。知道了原来好几千行的 configure 脚本都不是人写的，怪不得我写不出来呢，呵呵。

虽然在家编程很顺利，但是一上班还是头大。这个大企业说是管理正规，但是也太有点官僚主义了。hub 坏了想买一个都这么费劲，组长批，部门经理批，财务批，折腾一套下来，半个月都过去了。有这功夫我自己拿电线焊一个都焊出来了。”

主人的 rubbish 金刚终于还是放到网上去了，幸运的是，网络上的众多牛人们，为他修改了很多问题，把他调教的规规矩矩的，也不浪费内存了，也不乱改文件了，也不死机了，腰不酸了，背不疼了，现在我们都开始喜欢这个家伙了，他好，我们也好～

这大概就是开源的力量吧。我们现在都不好意思叫他 rubbish 了，就直接叫金刚。金刚发布前的那段时间，我们几个软件都很纠结，担心主人运行他，担心系统被他搞坏。那时候主人也很纠结，纠结的是怎么能够把他发布到网上去。那时候主人整天研究 shell 编程的技术，天天对着那三万多行的 configure 代码，出神的看着，嘴里念叨着：“这是哪位神仙大姐写的脚本啊……，这么多行得写多长时间啊……”，

直到有一天，他终于将眼睛聚焦在了 configure 文件前面的那段注释中的一句话：

```
# Generated by Autoconf 2.52.20081225
```

主人顿时如醍醐灌顶一般，看着这一行注释，看着这个 Autoconf，心里反复的呼喊，声音越来越强烈，直到终于爆发，脱口而出：“靠！原来是用软件自动生成滴！！！”

顿悟之后主人立刻叫来狐狸，本着“内事不明问老婆，外事不明问 google”的宗旨，直奔 ww.google.com 而去。

一番查找后，主人终于大致了解了 autoconf 这个软件。咱说 gcc 他们就像施工队，make 就是包工头，configure 就是分析师，那这个 autoconf 大概就是市政规划局了。有了他，什么 Makefile，configure 脚本，全都不用自己写，都由他一手代办。规划局的工作，就是根据源代码的结构和组成，来决定如何根据环境因地制宜，就地取材的施工，最终派出一个专门的分析师——也就是 configure。之后在安装的时候 configure 就可以根据目标系统的环境以及既定的几套施工方案，来写出 Makefile，再交给那里的 make 去指导施工。虽然软件叫做 Autoconf，但其实并不是只有他一个人。既然叫做规划局嘛，那就不可能是一个人，你见过哪地方的规划局就一个局长了？我们的规划局成员有四个：aclocal、autoconf、automake、autoscan。你要想自动创建 makefile 和 configure 脚本，就得跟这哥四个说。虽然可能你已经有 makefile 了，只是缺少 configure，但那不行，他们向来是买一送一，搭配销售。你写的 makefile 是没用的，必须得用他们创作的 configure 和 makefile，这个具体咱们待会再说。这四个人各司其职，其中，autoscan 是负责初步审查项目的。你的工程图纸画好了，得先拿给他看。他看了一遍之后，会给你写个报告。怎么还写报告？当然了，规划局嘛，审批个这处理个那的，不都是部门之间报告来，报告去的么。autoscan 写的这个报告叫做 confiugre.scan。

然后你要做的就是拿着这个报告去找 aclocal 审批。但是如果你真的直接拿着 autoscan 写出来的报告去，准过不了。autoscan 说是初审，但其实他本事一般，图纸都不一定看的懂。（那为啥他能在这规划局干活？听说主要因为他爸是啥刚。）他写出来的报告一般驴唇不对马嘴，你还得动手改改。改过的报告还得改个名字，叫做 configure.in。aclocal 也知道 autoscan 的那点事，都是圈里混的，谁还不明白谁阿。拿过 configure.in 来就知道，这是你改过的了，那就好好看看。看完了就批了么？一看你就没跟规划局干过，这才哪到哪阿。看完了他也会写个报告，叫做 aclocal.m4，之后你就要拿着这两个报告找到 autoconf。

autoconf 就是专门负责指派分析师的了。他看了两份报告后，一般会沉思一会，说说目前如何困难，人手不足之类的话，最终在你一再的苦苦哀求以及威逼利诱之下（一般威逼的不多），无可奈何的说：好，就给你派个分析师吧！于是，一个 configure 脚本诞生了。但这就完了么？当然没那么简单，这个 configure 是有条件的，他必须搭配规划局制定的 Makefile.in 才能工作。

有人问了，这个 Makefile.in 是什么阿？我已经有了 Makefile 还要他干什么？咱不是说了么，你的那个 Makefile，甭管写的多么的天花乱坠，也是白搭，人家规划局派出来的 configure 根本都不会瞧上一眼，人家 configure 要写自己的 Makefile 来用。你又得说了，那你

这 configure 就赶快写出来自己的 Makefile 阿。你看你，不讲道理了不是？这 makefile 那么复杂，哪能就这么凭空写出来阿，总得有个参考，有个蓝本，有个全市统一 Makefile 模板之类的吧。这个模板，就是 Makefile.in。

那么这个文件从哪来呢？嘿嘿，不是还有个 automake 没出场呢么？这回你该求他了。automake 就是专门写 configure 需要的 Makefile.in 的，您看咱规划局给您搭配的多好。那么 automake 直接就能写 makefile.in 么？也不是，人家比较忙，这个您也得理解，局里那么多事情，就算没啥正经事，也少不了大家一起出去会个餐，考个察什么的。就算不会餐不考察，谁也免不了得上个网，偷个菜，斗个地主扫个雷啥的吧？所以呢，automake 是没功夫从头给你写出一份 makefile.in 的，你得先给 automake 写好一个框架，然后人家才好动笔。这个框架呢，就叫做 Makefile.am。有了这个框架，交给 automake，他就可以给你写出 Makefile.in 了。之后就可以把这些文件连同源码一起打包发布了，用户拿到软件包之后只要：

```
./configure
make
make install
```

就可以把软件安装上了。

刚才说到哪了？哦对，主人终于闹明白了，原来他一直说的那个简直不是人写出来的 configure 脚本还真不是人写出来的。于是上网学习 autoconf 的用法，经过一番辛苦的学习，他终于理清了头绪，开始动手了。

只见主人来到存放金刚源码的那个目录，目录里面现在有 main.c、board.c、ai.c, board.h、ai.h 几个文件。金刚的功能也被主人删除的只剩下跟主人下棋这么一个功能——因为就这个功能还算有点用。来到目录之后，主人先是叫来 autoscan 来进行扫描。咱不是说了 autoscan 是初审么。只见 autoscan 同学大摇大摆来到主人的金刚源码目录，东瞅瞅，西看看，挨个打开每一个文件，终于搞清楚了各个文件之间的关系，然后按照一套很官方的格式，写了初步审核报告书，就是那个 configure.scan，之后就会去睡觉去了。主人拿到报告书，当然知道，这只是万里长征，才走完了第一步。赶紧叫来 gedit 小弟，修改报告书。把一些完全不着边际的东西删掉，或者注释掉（也就是在那行的前面加上 # 号。）修改后的报告书是这个样子的：

```
#
-*- Autoconf -*-
# Process this file with autoconf to produce a configure script.

#AC_PREREQ([2.64])
AC_INIT(main.c)
AM_INIT_AUTOMAKE(all_in_one, 1.0)
#AC_CONFIG_SRCDIR([ai.c])
#AC_CONFIG_HEADERS([config.h])
```

```
# Checks for programs.
AC_PROG_CC

# Checks for libraries.

# Checks for header files.
#AC_CHECK_HEADERS([stdlib.h])

# Checks for typedefs, structures, and compiler characteristics.

# Checks for library functions.

AC_OUTPUT(Makefile)
```

看着挺多，其实都是注释，真正有用的就这么几行：

`AC_INIT(main.c)` ——这句是说明这个工程的主要图纸是那个文件。

`AM_INIT_AUTOMAKE(all_in_one,1.0)` ——这行是汇报这个项目的名称，叫做 all_in_one（大概是一锅烩的意思。），版本是 1.0 版。

`AC_PROG_CC` ——这句是说，最终的 configure 需要检查 c 语言编译器是否正常。

`AC_OUTPUT(Makefile)` ——这行是说明，最终的 configure 需要产生的文件，叫做 Makefile。

其他的，都是废话。

主人把改好的报告改名为 configure.in，然后叫醒了 aclocal 来看报告。aclocal 不紧不慢的起床，伸着懒腰走进工作间，拿起桌上的茶水杯，掀开杯盖，撇一撇浮在水面上的茶叶，拿嘴吹一吹，喝一口，盖上杯盖，放下杯子，开始看报告。扫了两眼后，就知道是怎么回事了，也不用主人多说话，直接写了一份复查报告，叫做 aclocal.m4，扔给了主人，就赶紧回去继续睡觉去了。要说人家 aclocal 的办事效率那还真是不错，毕竟人家写的 aclocal.m4 不用主人修改，直接就可以往上交了。

主要领导终于出场了，主人赶紧又叫来 autoconf，让他指派分析师 configure。autoconf 这回倒是没太耽搁，看了看两份报告后，就生成了 configure 脚本。

那么这个脚本得搭配他们专门的 Makefile 蓝本阿，所以主人又找来 automake。让她写，automake 拉着长声说：“这～个～～我们这里～工作也比较忙啊，是不是？……要按说呢，这个文件我是应该给你写滴～～不过我们这里每天这么多人来～～我要是一个一个写呢～哪天才能写完啊？同志啊，为了帮助我们提高办事效率，也为了你自己早点拿到 Makefile 蓝本，更为了我们祖国的三个现代化，诶不对，五个，也不对，几个来着？咳，甭管多少个吧，反正呢，您是不是自己先写个草稿给我，我也好帮你赶快写出蓝本呀。”主人听得都快扔板砖了，心说不就你想犯懒这么点事么，至于跟我撒这么半天么。赶紧动手写草稿，这个草稿叫做 Makefile.am，内容很简单：

```
AUTOMAKE_OPTIONS=foreign
bin_PROGRAMS=all_in_one
all_in_one_SOURCES=main.c  ai.c  board.c
```

第一行呢，是行业规定，就这么写。（其实这是 automake 的选项啦。Automake 主要是帮助开发 GNU 软件的人员来维护软件，所以在执行 automake 时，会检查目录下是否存在标准 GNU 软件中应具备的文件，例如 'NEWS'、'AUTHOR'、 'ChangeLog' 等文件。设置为 foreign，automake 就不会查这些了。）

第二行吧，就是说明编译之后的程序应该叫做 all_in_one。

第三行嘞，就是说这个工程包括 main.c ai.c board.c 这三个文件。

就这么简单，草稿就写完了，之后再把 automake 叫出来，总算是给写出了 makefile 的蓝本——Makefile.in。

做完了这些之后，这个工程就可以打包发布了。用户拿到这个包，解开之后，就直接 ./configure ;make ;make install 就把软件安装上了。主人欣喜的看着自己整出的这个像模像样的软件，看看 configure 脚本，四千多行！心想：不知道的人看见在这个脚本一定以为我是大牛呢吧，哈哈哈～

再运行一下 configure，看看生成的 Makefile，五百多行，哈哈，隐然感觉自己已经成为高手了一样～

于是，在主人的 YY 中，我们结束了那一天的工作 

Chapter 6

第六章

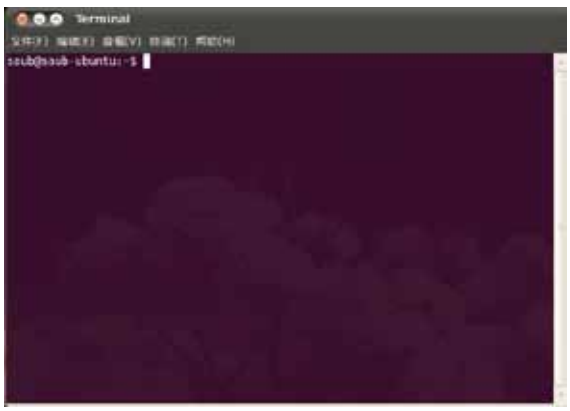
shell

6.1 这就是 Shell

能够有一个理解你，信任你的主人，对于一个 ubuntu 系统来说是一件非常幸福的事情。

虽然我们 linux 的图形界面已经比较先进了，绝大多数操作都完全可以用图形界面来完成。但是就像吃过麦当劳肯德基不等于吃过西餐一样，连刀叉都不知道怎么用的你也好意思说你会吃西餐？连终端都没进去过你同样也不好意思说你会用 Linux 吧？或许有人不是很赞同这个说法，不过我的主人是这么认为的。主人现在在周围的同学同事里面已经俨然是一位 linux 大牛了，能够把这个别人见都没见过的系统耍的上下翻飞（比如 3D 桌面转屏幕的时候），真的是很拉风的。可是他并没有停留在这种表面上的绚丽。如果仅仅会用图形界面，不会几行命令，那作为一个号称是会用 linux 的大牛来说是肯定很没面子的，所以主人为了进一步巩固 linux 大牛的地位，决定去学习 linux 的命令行。

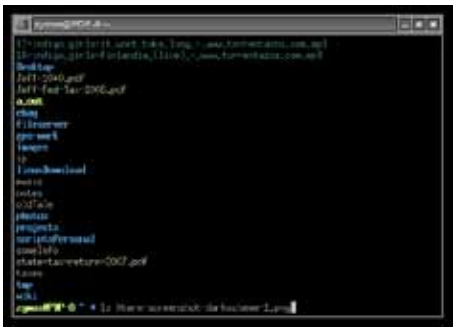
命令行并不神秘，打开应用程序 →附件 →终端，就看到了。

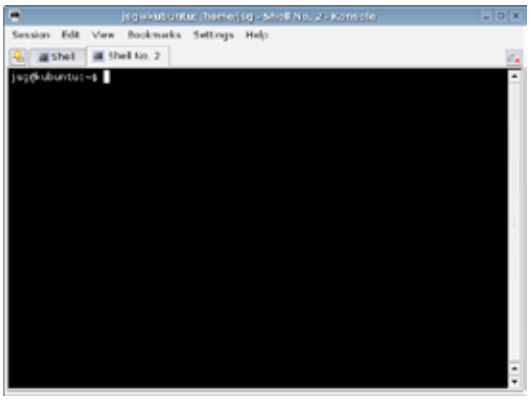


在这里，你可以近距离的跟我交流。我们操作系统是很希望用户能够和我们使用命令行来交流的，向朋友般倾诉，感受彼此的心声，而且还低碳环保，节省能源（省 cpu 和内存啊！）。用户就相当于老板，使用图形界面的用户和使用字符界面的用户是两种完全不同的老板。前者高高在上，拒人千里之外，就会比划，有事都不直说。比如他用手一指桌面上的文件，冲你“恩～”一声，你就得明白，他是想让你

选中这个文件，你就赶紧过去把文件拿起来。他要是指着文件冲你“恩，恩～”两声，你就得明白他是让你打开文件。（这里你还得注意，他“恩”的那两声之间的间隔长短，要把握好分寸。要是间隔很短，那叫“双恩”，要是间隔长，那叫“恩”两下，对应的操作是不一样的。）

他指着电灯开关的按钮冲你“恩”一下，那就是让你过去把电灯打开，他要是用手冲着桌面上的一堆东西指一圈，然后嘴里拉长声的“恩～～～～～～～～”一下，那就是让你把这些东西全都选中。给这种老板打工，需要很大忍耐力。要是遇到第二种老板就要好的多，这种老板，平易近人，没有架子，有事会跟你平等的交流：小笨啊，你列个表，看看我这目录下都有什么东西啊；小兔啊，3 分钟后下班啦，到时候记得关机啊；小笨啊，咱挺寂寞的，放个片子看看吧，就放那个 wall-E.avi 吧。这样的老板，跟你工作起来，那才是心旷神怡。用命令行的用户，就相当于这样的老板，舒服。



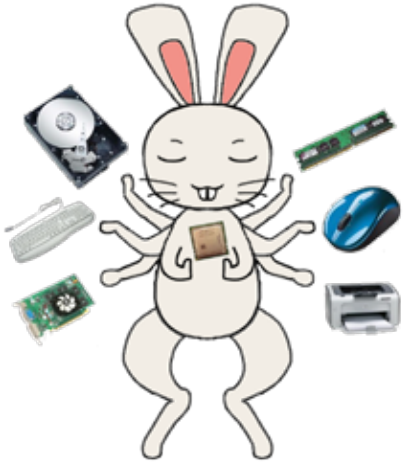


看到主人用鼠标指着终端，“恩”了一声，我就知道，该去把 gnome 终端叫起来了。gnome 终端就像查皮那里的命令提示符一样，是个虚拟终端，不过比那个可先进多了。所谓虚拟终端，是相对于真正的，黑漆漆的，按 alt-ctrl-f1 出来的那个终端来说的。虚拟终端的工作是在桌面上申请一个窗口，然后在里面模拟显示出一个黑漆漆的界面。（其实仔细看 gnome 终端还算漂亮啦，还能背景半透明呢。）除了 gnome 终端之外，还有很多用来输入命令的终端，像 konsole、rxvt、xterm 等等，他们都是虚拟终端软件。你可以多装几个虚拟终端对比对比，看哪个好用。反正无外乎都是一些黑漆漆，白惨惨，绿幽幽的窗口，在里面可以输入命令而已。

有的人可能会有顾虑：虚拟终端，那是不是里面执行的命令不是真正的 linux 命令呢？这些不一样的虚拟终端对命令的执行结果都一样么？

不一样？错！一样？还是错！

他们根本就不会执行命令，他们只是个界面而已，都是浮云。真正处理命令的不是他们，是 shell。在里面运行的命令并不归虚拟终端软件去解析，而是找到专门的命令行的程序，这个程序就叫做 bash，这个咱们待会再说。



那这个 shell 又是个什么东西呢？

shell，顾名思义，就是个壳！怎么叫壳呢？咱们慢慢说。我们 linux 是个内核，我这个内核是可以做很多事情的，整个电脑的硬件都归我管。我和硬件的关系，大概就是这样：

很威武吧。好，我现在管着所有硬件，那应该用这些硬件干点什么呢？我不知道，因为没有人给我下命令啊。这下命令的就是人类用户，就像我的主人。那么主人来了之后，大概就是鸡同鸭讲了。

哎，没办法。你们人类那么多种语言，我……，我却是一句都听不懂啊！你们人类呢？也一样，就是听不懂我说的机器语言，那怎么办？就得有个翻译，同声传译嘛。有了之后就好多了。

看，这个 shell 的作用是不是就像个壳子罩着我，让我能够和主人更好的交流呢？ shell 的名字就是这么来的。他是我这个内核用来和他们人类交流的一个外壳。

shell 的作用我们知道了，那么他的本质呢？就是个 /bin/ 目录下的一个二进制程序。比如我们 ubuntu，用户默认的 shell 是 bash，就是 /bin/bash 这个二进制文件。当主人打开终端的时候，终端程序（比如我们这里是 gnome terminal，为了省事，以后我我们就叫他 G 终端吧。）就会进入工作间，先绘制出一个界面，（当然，这需要图形部门的配合），然后就去找你这个用户的 shell 程序。那么用户的 shell 程序是什么呢？这个在 /etc/passwd 文件里有记载，G 终端找到 passwd 文件里对应当前用户的一行，比如我们这个就是这样：

```
lanwoniu:x:1000:1000:lanwoniu,,,:/home/lanwoniu:/bin/bash
```

这一行的最后一段就说明了这个用户的默认 shell 是 /bin/bash。于是 G 终端就去叫醒 bash，bash 起床，通过 G 终端来跟主人用文字交流。那么最先说的一句话大约就是“你好”，当然不会这么不专业，这句话用 bash 专业的语言说出来就是这样：

```
lanwoniu@lanwoniu-ubuntu:~$
```

这行是什么意思呢？首先最前面的，也就是 @ 之前，是当前用户名称。@ 后面是计算机名，这两个都好理解。“：”后边是当前所在目录，就是当前输入命令的人所在的位置。”~”代表用户的家目录，也就是 /home/< 用户名 > 这个位置。$ 则是命令提示符，$ 后面，就可以输入命令了。顺便说说，普通用户的提示符是 $，root 的提示符是 #，不过我们 ubuntu 不能用 root 登录，所以基本你也看不到 # 提示符了。

正说着，主人已经开始敲命令玩了。什么 ls，free，top，fdisk 等等，挨个试验。于是工

作间里也开始忙碌了起来。你可能会以为 bash 会在主人的指挥下跑来跑去，执行各种操作？呵呵，其实完全不是那么回事，bash 只是作为一个命令的传达者而已，真正干活的是那些命令们。也就是 ls，free 这些家伙。这些所谓的命令，其实都是一个一个的小程序，或者说一个个小的软件而已。就跟狐狸妹妹，OO 老先生一样，只不过比他们小巧很多。如果你愿意，也可以把 firefox 看作一个上网用的图形化界面的命令，为了好说，咱们以后管这些家伙叫做命令程序吧。当用户输入命令，比如 ls 的时候。首先 ls 这两个字符是被传给 bash 的，bash 怎么处理呢？首先 bash 要看输入的字符是不是自己的什么关键字，比如 for、history 之类的，如果是的话，就归 bash 来处理了，如果不是，那就说明主人是要找个命令程序，bash 就要负责去找到主人想要的这个程序，并且叫他起床干活。那 bash 去哪里找呢？不知道你知不知道有个叫做环境变量的东西？跟 windows 的那个环境变量差不多，其中有个环境变量叫做 PATH，这里面记录着 bash 去找程序的路径。如果你想看看 PATH 到底是什么，运行 echo $PATH 就可以了。会得到类似这样的输出：

```
lanwoniu@lanwoniu-ubuntu:~$  echo  $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
```

当主人运行一个命令，比如 ls，bash 就对照着 PATH 里面的设置，先去 /usr/local/sbin 房间里面找，敲开门，客气的问：请问 ls 是住这屋么？ 等了半天，除了墙角的蜘蛛网上那八条腿的小家伙寂寞的弹了几下琴弦之外，再也没有活物给他任何回应，于是 bash 意识到这屋没人，赶快去下一屋。到了 /usr/local/bin，依然是很礼貌的敲开门问候，里面只有一位主人前几天安装的叫做 Maya 的软件，只听那位叽里咕噜的说了几句 2012 啥的玛雅语，bash 也听不懂，不过反正他不是 ls 就对了，赶紧去下一间。推开 /usr/sbin，这回里面很热闹，而且都是重要人物——管用户创建的 useradd、每天启动必备的 gdm、负责跟通过网络跟查皮共享文件的 smbd 和 nmbd 等等。一听 bash 来找 ls，useradd 没好气的说：“呀，ls 怎么可能在我们这里呢？我们这里都是管理级的程序，那个 ls 是谁都能运行的，怎么会在 sbin 里？你得去 bin 里面找阿。”，bash 只好客气的退了出去，继续去找 /usr/bin,/sbin/, /bin，终于在 /bin 里面找到了 ls，于是赶紧叫醒 ls，让他去干活。至此，bash 的任务也就结束了，他就回去等待主人的下一个命令就好了。等到下一个命令来了，bash 就再挨个去各个屋里找主人要的命令。

需要注意的是，bash 是严格按照 $PATH 所记录的位置去找命令程序的，一般这个 PATH 里面并不包含当前目录，也就是“./”。所以 echo 并不会在当前目录查找命令程序。有的人可能当前在 /home/bob/bin/ 目录下，这个目录下有一个 kiss 程序，于是他直接运行 kiss，结果 bash 告诉他没找到这么个命令，他就很困惑，还找来 ls 来证明：这明明就在眼前的东西，你怎么就告诉我没有呢？

其实不是没有，是你没让 bash 在眼前这个目录找嘛。那么要想运行一个程序就必须把它放在 /bin,/usr/bin 之类的这些地方么？当然也不是，当你不告诉 bash 路径的时候，bash 是去 PATH 变量里找的，如果你自己知道你要运行的程序不在 PATH 变量所记载的目录里，那就直接了当的告诉 bash 你要的程序在哪，bash 就能找到了。比如你运行 /home/bob/bin/kiss 命令，就是告诉 bash，去 /home/bob/bin/ 目录下找一个叫做 kiss 的家伙出来干活。如果你像刚才说的

似的现在就在这个目录下，就可以把刚才那很长的地址简单描述为“当前目录下的 kiss 程序”，当然你跟 bash 说这句中国话是不可能的，那么当前目录怎么表示呢？很简单——“./” 所以运行当前目录下的 kiss 程序就运行“./kiss”就好了。说了这么多，是不是不会再有人误解“./”是运行的意思了呢？（记得 ./configure 吧，编译软件的三部曲之一，就是运行“当前目录下的 configure 脚本”的意思。）

主人敲打着 ifconfig 命令来查看网络链接状况，忽然想起这个命令应该还能改变自己的 IP，于是想试试，可是又忘了具体是怎么用的了，怎么办呢？ bash 告诉他：别着急，我给你找个人问问。这个人，纯爷们！然后 bash 扭头冲着硬盘里喊：Hey man，你的出来，说说这是怎么个意思的干活。随着咔咔嚓嚓一声硬盘响，只见内存中走来一人，人高马大，膀大腰圆，扇子面的身材，胳膊跟大腿一样粗，这就是 bash 说的那个 man。这纯爷们说出话来如同打个炸雷一样：嘿！你好啊。洒家我是专职命令解说员，你有什么想知道的吗？

主人输入了命令 man ifconfig。意思就是问 man，这个 ifconfig 是怎么用法。man 仰天大笑一声：“嚯哈哈哈，要说这个 ifconfig 嘛……，不难，听我慢慢地道来！”

既然说到了 ifconfig，那就说说他的用法。这个命令是用来配置网络的。如果你想查看现在的网络链接状况，就找他就好了。只要运行 ifconfig，就可以得到类似如下的内容：

```
lanwoniu@lanwoniu-ubuntu:~$ ifconfig
eth0      Link encap:以太网  硬件地址 00:1f:d0:8b:9c:10
          inet 地址:192.168.1.100 广播:192.168.1.255 掩码:255.255.255.0
          inet6 地址: fe80::21f:d0ff:fe8b:9c10/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:34303 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:26207 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:40115963 (40.1 MB)  发送字节:2356235 (2.3 MB)
          中断:29 基本地址:0x4000

lo        Link encap:本地环回
          inet 地址:127.0.0.1  掩码:255.0.0.0
          inet6 地址: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  跃点数:1
          接收数据包:215 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:215 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:0
     接收字节:15116 (15.1 KB)  发送字节:15116 (15.1 KB)
```

基本都是中文，也就不用解释了，如果不知道什么是 ip 地址，不知道什么是掩码的，那先补习一下网络基础。如果你想设置网卡的 IP 之类的呢？那也找 ifconfig，很方便，比如要修改 eth0 网卡的 IP：

```
lanwoniu@lanwoniu-ubuntu:~$ ifconfig eth0 192.168.1.100
```

就可以了，eth0 后面就是你要设置的 ip 了，如果同时想设置下子网掩码，就在后面加上一段，类似这样：

```
lanwoniu@lanwoniu-ubuntu:~$ ifconfig eth0 192.168.1.100 netmask 255.255.255.0
```

经常有人想修改网卡的 mac 地址，在查皮那里好像还要改什么注册表什么的，我们这里还是交给 ifconfig 就可以了，简单：

```
lanwoniu@lanwoniu-ubuntu:~$ ifconfig eth0 hw ether xx:xx:xx:xx:xx:xx
```

其中 xxx 就是你要改的 mac 地址。最后，如果你看着某个网卡不顺眼，可以停掉他，就这样：

```
lanwoniu@lanwoniu-ubuntu:~$ifconfig eth0 down
```

如果过会你又怀念他了，还可以把它找回来：

```
lanwoniu@lanwoniu-ubuntu:~$ifconfig eth0 up
```

说了这么多 ifconfig 的用法，最后还是要说明一下，ifconfig 这家伙做事还是有些毛糙，他只管立即生效的东西，所有你做的设置在系统重启之后就算都失效了。

故事外的事——bash 的历史

在最初创造 UNIX 系统的时候，Ken Thompson 大牛在 Bell 实验室写了一个简单的程序，将它作为新的 UNIX 操作系统的接口界面来让系统和人类进行交流。这个程序就叫做 shell，那时候还没正式起名字，为了让使用这个 shell 的人们不忘记作者的辛劳，大家也就管它叫 Thompson shell 了。Thompson shell 的功能很简单，用户通过它输入一些指定的命令，它负责解释为需要计算机做的操作，并去执行。另外它还能够支持一些简单的脚本，就是把一堆命令写进一个文件里依次执行。但并没有更高级的例如流程控制，分支，变量，函数之类的东西。后来这个 shell 进行了扩展……

同时，Ken Thompson 的同事，同在 Bell 实验室的 Steve Bourne 也写了一个 shell 程序，作为一个有很多重要程序要写的大牛，Steve Bourne 也没来得及给他写的 shell 起名字，于是人们同样为了不忘记作者的共享，也就管这个 shell 叫做 Bourne Shell。这个 shell 跟

Thompson shell 不同，他加入了流控制，可以写简单地函数。等到了 1970 年代晚期，每种 shell 在 Bell 实验室都有不少的人用，于是形成了两个派别。说来这两个 shell 应该是都很不错的，用熟悉了都很顺手，但是这两个 shell 是不兼容的，就如同修习了少林的内功后再去学武当的心法，多半不是很习惯。但是作为一个成熟的商业化的系统，总该有个默认的，标准的 shell 才好方便用户学习使用。于是在 bell 实验里，分别支持 Thompson shell 和 Bourne Shell 的两大帮派，进行了激烈的辩论，经过三次连续的 Unix 用户组集会上两大帮派的斗争之后，终于确立了 Bourne shell 为了 Unix 的标准 shell。

等到 1978 年，Bourne Shell 随着 Version7 Unix 一同发布，于告别了实验室，和广大用户见面了。9 年后，1987 年，一个叫做 Brian Fox 的家伙非常喜欢 Bourne Shell 并且觉得它还可以更加完善，于是就开始在 Bourne shell 的基础上进行创造，几年后成为了一个更加完整而且好用的 shell。出于对 Bourne shell 的缅怀和崇拜，他将这个 shell 命名为 Bourne Again Shell——简称 bash。现在，bash 是绝大多数 linux 系统，以及 Mac OS X v10.4 系统的默认 shell。甚至还被移植到了 Windows 系统上，什么？你没见过？那你听说过 Cygwin 吧？哈哈，那里面就是 bash 啦。

6.2 这么用 Shell

于是，主人那敲敲打打的命令行生活就这样开始了。从那以后，工作间里少见了红酒大师；看不到盒子妹妹；心有灵犀也不常来上班了；OO 老先生也难得起床了，工作间里净是些命令行的小程序在跑来跑去。唯一还经常出现的熟悉的面孔就是狐狸妹妹了，因为主人总要上网去查查命令行的相关知识，毕竟跟那个彪悍的 man 说话还是有点费劲的。

后来主人觉得玩虚拟终端还不够过瘾，现在每次都要按 ctrl-alt-f1 进入真正的终端来敲名令，看着一屏屏的字符好象很有成就感的样子，如果遇到需要查的东西再按 ctrl-alt-f7 回图形界面来查。

进入终端以后，最先被主人叫起来的是 ls，这家伙算是出镜率最高的一个命令程序了，ls 是 list 的缩写，他是专业负责清点物资的。用起来也方便，就敲 ls 就可以了，他就会把当前目录下的所有文件给你列出来。如果你觉得光看文件名不够的话，还可以这样：“ls -l ”，这样以列表方式查看，信息就更丰富了。主人运行了一下 ls，看到了当前目录下的所有文件。所谓当前目录，就是用户现在所在的目录，比如你在你家卧室发呆，那么你的当前目录就是卧室。过一会你又去客厅发呆了，那当前目录就是客厅，然后你去厕所发呆，当前目录就是厕所（怎么到哪都发呆……(⊙ _ ⊙)?）

那么主人现在的当前目录是哪个目录呢？就是他的家目录，就是图形界面的“位置”下的“主文件夹”，也就是 /home/lanwoniu 这个目录。当主人每次打开终端的时候，无论是虚拟终端还是 ctrl-alt-f1 进入的终端，刚一进去，都是在主人的这个家目录里。

不过毕竟不是什么事情都要在家目录里做的，如果你在这个目录里看够了，想出去走走，到其他的目录逛逛，这时候就需要 cd 了。

cd 不是那个光盘，而是 Change Directory（改变目录）的缩写。这个命令可以改变当前的目录，他就像出租车一样，可以让你到达你想去的任何一个目录。（当

然，前提是你得有权限）用法就是：

```
cd  < 后面写你想去的目录 >
```

那么，如何描述你想去的目录呢？一般有两种方法：绝对路径和相对路径。

绝对路径，就是无论去哪都从根上说，比如你打车，司机师傅问你去哪，你说：“我去地球，亚洲，中国，北京市，崇文区，羊肉大街，排骨胡同，376 号。”这么说就是绝对路径，无论什么地方，都从一个根本的位置上说起，（比如地球……当然，你愿意从太阳系说也行。）层层递进，最终说到最小最详细的那个地方为止。那我们这里的目录的地址当然不能从地球开始说了，我们的根目录“/”就是那个最初的，根本的位置，无论你去哪个目录，都可以从这里说起，比如 cd /usr/share/system/hidden/av，这就是说：要去根目录下的 usr 目录下的 share 目录下的 system 目录下的 hidden 目录下的 av 目录。（埋藏的真深啊！）

这么说很准确并且直接，不过有时候也比较费劲。所以，cd 还可以支持相对路径。

相对路径，就相当于你打车，司机师傅问你去哪，你说：“就前面那路口，左转，过仨红绿灯走 700 米有一精神病院，到时候我指给您，您就靠边停车就行了。”这种描述方法就是以当前所在的地点为起始地点进行描述，而不用从外太空开始说了。那么具体到我们这个系统里怎么说呢，就这样 cd abc/bcd/dca/cbd/adb/av ，这句话的意思就是：要去当前目录下的 abc 目录下的 bcd 目录下的 dca 目录下的 cbd 目录下的 adb 目录下的 av 目录。（归根结底还是 av 目录……）

主人学会了 cd 命令，兴奋的在终端里 cd 来，cd 去的，像个跑进游乐园的小孩子，到处走到处看。看着看着，忽然觉得有点迷路了。静下来想一想：我现在是在那个目录呢？是在 /usr/bin，还是 /bin，还是 /usr/local/bin 呢？其实，他看一眼那个提示符 $ 前面的内容就可以了，默认设置下这里显示的就是用户所在的目录的绝对路径。不过这个提示符的格式是可以修改的，如果修改了，这里显示的不是当前的路径了，那怎么办呢？我们早为您想到了，命令行中配有专业的引导员，告诉您您现在所在的位置，这位引导员就是 pwd。

可别一看名字就以为他是负责修改密码的阿，其实他跟 password 一点关系都没有，他是 Print Working Directory 的缩写。用法简单，就输入 pwd 就行了，他就会告诉你现在所在的目录。

忽然，主人发现开机了半天还没有联网呢，这个机器上有个无线网卡，需要链接一下家里的无线路由器才可以上网，在图形界面有网络管理器负责，在字符界面下，也同样有称职的人员为你服务，这就是 iwconfig。主人又叫来 man 来解释解释 iwconfig，纯爷们发话：“嚯哈哈哈，这个 iwconfig 嘛，就是如此……比如……你这样……对啦……然后……没错，嘿嘿，果然……就连上啦。”您问到底怎么样？您自己去问 man 吧，我在这也不好把 iwconfig 的手册翻译一遍不是。学习了一下之后，主人敲 iwconfig 来试着联网。可是，他忽然觉得，这个 iwconfig 字母有点多，不如 ls，cd 那样简短。这么多字母输入起来有点麻烦，有没有什么省事的办法呢？

很多人不喜欢键盘，不喜欢打字。其实想想，早在电脑刚刚被发明出来的时候，键盘就已经是每一台电脑所必备的输入设备。作为从那个字符界面的时代走过来的我们 Linux 系统，自然充分考虑的通过键盘操作整个系统的便捷和效率问题。直到现在，使用键盘操作 linux 都会拥有意想不到的高效率和成就感。

我以前很不明白，命令键盘可以发送上百个命令，用起来应该很方便才对，为什么人类就那么喜欢那个只能发送：向上，向下，向左，向右，左键，右键，这六个命令的鼠标呢？（当然，现在的鼠标还多了滚轮，还有的鼠标有更多的按键，但是那也比键盘少啊。）后来见多识广的 OOo 老先生给我解释，我才明白，原来是因为人类记忆力不行，没有我们软件这么可靠。记不住那么多个键，于是只好用那只能发送六个命令的鼠标了。

好了，绕的有点远，其实说起来在我们的命令行里通过键盘敲命令是很方便的，只是很多人不大熟悉如何节省时间而已，都以为用键盘和我交流跟用键盘和那个剁死系统交流一样麻烦呢。其实我已经很人性化了，就因为键盘上有个键——Tab。

> 看一个人的键盘，就可以猜测出他平时用电脑干什么。如果 W,A,S,D,U,I,J,K 严重磨损，说明这哥们玩拳皇的；如果 A，Shift，Ctrl，1，2，3，4……9，0 严重磨损，说明是个玩即时战略的，星际魔兽之类；如果 ALT,S 或 Ctrl,Enter 磨损，大概是天天聊 QQ；如果 Tab 键严重磨损，那估计就是个 Linux 高手了。

在 Linux 的命令行下，Tab 键起着命令补全的作用。比如说，你要运行 ifconfig 命令，你可以不用完全输入这 8 个字母，只要输入 ifc，然后按 Tab 键，Bash 就知道你要干什么了，因为所有可以运行的命令里面以 ifc 开头的就只有 ifconfig，所以当你按下 Tab 键的时候，他就会替你写出完整的命令：ifconfig。这都因为在你按下 Tab 键的时候，Bash 会去 PATH 变量所设置的所有目录里遍历一遍，检查了里面所有的有 x 权限的文件，结果只查到了 ifconfig 文件（命令其实就是个可执行文件）。

之所以这么快，是因为他早就把这些重要的东西缓冲进内存了，所以下次就别抱怨我们 linux 动不动就把你内存占满了哦。那如果你再少写个字母呢？比如你只写了 if，然后就按 Tab 键，Bash 遍历了一边 PATH 中的路径后发现，有 4 个命令是以 if 开头的，所以他不知道你要的是哪个命令，于是就不做任何动作。这时候如果你再按一下 Tab，他就会提示你：以 if 开头的命令有 if、ifconfig、ifup、ifdown。然后你自己看需要的是哪个，照着输入就行了，很交互是吧？这样除了减少按键次数以外，还有一个好处就是你可以不必完全记住整个命令，能够记住前几个字母就可以通过 Tab 把整个命令回忆出来。

有了 Tab，就让用户输入新命令的时候省事了不少，还有一个 history 功能，可以让用户重复以前输入过的命令的时候省心。如果你想输入上一次输入的命令，就按一下向上箭头，就看到了；如果想要再上一次的命令，就再按一下；如果想要再再上一次的命令，就再再按一下；如果想要再再再上一次的命令，就再再再按一下；如果想……如果想写作文的时候凑字数，你就跟我学。好了，总之是可以通过方向键选择以前运行过的命令，如果

想查看很久远以前的命令呢？也可以，输入 history。这是一个命令，可以显示之前运行过的 n 条命令，默认情况下 n=1000，现在图形界面越来越发达，输入命令的机会越来越少，估计 1000 条都能把去年的命令显示出来了。说起来 history 命令也没啥神奇，他之所以能够显示曾经运行过的命令，不是因为他有啥水晶球，而是负责解释主人命令的 shell 会把每一条命令记录下来，就写在 ~/.bash_history 文件中。history 只不过是把这个文件打开，显示出里面的内容罢了。

主人输入了 iwc 三个字母，然后按了一下 Tab 键，bash 赶紧给他把命令补全，iwc 变成了 iwconfig，主人直接按下回车，哈哈，果然省事了不少。只简单的运行 iwconfig 的话，只是列出现在的无线网卡的状态，要想了解具体的使用方法——对了，还得问那个爷们，于是一阵“嚯哈哈哈”的笑声又传来了。

主人经过学习，用 iwconfig 连接到了无线路由器，但这只是相当于拿个网线插到了路由器上，还需要有一个命令来配置网口的 ip 地址啥的，对！就是 ifconfig 命令，之前主人在图形界面下就用过了。那时候主人运行 ifconfig，出了好多字，滚屏滚上去了，主人拖着虚拟中端的滚动条看了一下，这个动作如此的自然，没有任何异样的感觉，然而现在到了真正的黑漆漆的终端里，主人又随手运行了 ifconfig，忽然发现一个问题——滚上去的信息怎么看啊！

没有滚动条啊！！有木有滚动条啊！！！有木有啊！！！！有木有！！！！！

好，我们让主人自己咆哮一会去，咆哮的主人我伤不起阿。

其实就是想向上滚屏嘛，很简单，有快捷键，Shift+PageUP 是向上翻页，Shift+PageDown 是向下翻页，就这样。当然，可能光滚屏还不够，有时候想把命令输出的东西翻来覆去慢慢看，有什么办法没？肯定是有的，当初的前辈们都是生活在命令行下的，没事编译个软件啥的，那输出的提示动不动就好几百行（代码写的 warning 太多 -_-b），一屏是根本显示不下的，就算 80 寸！竖着放也不够！

显示不下了，后面的内容就会把前面的内容“顶”上去，“楼主”固然是看不见了，什么“沙发”、“板凳”、”地板”、”下水道”的也一样没希望，只能看到最后那几十条。那想看前面的怎么办？虽然可以用 shift+page up 来向上翻页，但一来向上翻的页数也是有限的，二来这也麻烦，一般人都是习惯从上往下看的，倒着往上翻就别扭了。那怎么办呢？这时候 more 就该出场了。

more 的功能就是分页显示，把所有要输出的内容先显示出一屏来，等着用户按回车，之后再显示第二屏，直到显示完全部内容。当然，用户也可以不等显示完全部就中途按 q 退出。那怎么用呢？就比如刚才主人这种情况，那就运行“ifconfig | more”。“|”符号叫做管道符，就在你键盘上的”\”键。这个符号的意思就是把前面一个命令的输出内容交给后面一个命令作为数据输入。于是，ifconfig 命令输出的那些个字符就像一大堆泔水一样哗啦啦的都流倒 more 这里了（怎么偏偏是泔水……），more 就把这些东西都先整的大桶缓存起来，然后先盛出一碗来给你看，“您看有没有想吃的？”，你看完了之后，他再去盛第二碗，第三碗……直到你把整个一大桶泔水都检阅完了，more 才结束工作。（当然，你要是坚持不到最后就吐了，那就按 q 退出）有了 more 就满足了么？不，还有他的死对头，less。

less 实现的功能和 more 基本一样，也是用来分屏输出的，同行是冤家嘛。不同的是，more 只能一页一页往下看，看完了就退出。less 可以上下翻页，看过去的东西可以按向上键或者 page up 键翻回去看。是不是比 more 更人性一点？另外，都看完了之后 less 是不会自动退出的，一定要按 q 退出。顺便说一下，more 和 less 不光可以通过管道将其他命令的输出当作输入，同时也可以直接查看文件。只要“more /< 路径 >/ 文件名”或者“less /< 路径 >/ 文件名”就可以查看文件内容，当然，只能是文本文件（里面是文本就行，不一定非得以 .txt 为扩展名，我是 linux，没有某些 OS 那么傻）。

通过这两个命令，您大概可以感受到我们 Linux 系统和有点软公司的系统的不同理念。用过剁死系统的都知道里面有个 DIR 命令，和我们的 ls 一样，都是显示文件用的。当文件很多的时候，dir 命令有专门的参数可以实现分屏显示，而 ls 命令就没有，只能一下子显示出来。为什么？因为分屏显示的事情是由 more 和 less 负责的，完全可以通过“ls | more”这样的组合实现分屏显示。linux 的理念每个程序只专注于一种功能的实现，而通过多个程序的组合可以实现任何功能。试想如果没有 more 和 less，ls 要负责分屏显示的话，那 history 命令是不是也要处理分屏显示的问题呢？那么所有输出行比较多的命令都要自己负责分屏显示，这些命令的源代码中都要有负责分屏显示的部分，这是一种无谓的重复劳动，而且各自分别实现分屏显示，效果多半也不一样，可能有的命令是按空格显示下一行，有的是按 n 显示下一行等等。与其这样不如把相同的功能独立出来，成为一个统一的，单独的命令。

好了，主人终于结束了咆哮，回到图形界面找狐狸妹妹去查怎么在命令行下翻页了。

说起命令行下省事的办法，还有个事情不能不提，这就是通配符。

过去用过“剁死”系统的同学可能知道，通配符就是“*”和“?”，“*”可以代表任意多个任意的字符，“?”代表任意一个字符。不过“剁死”系统下通配符的实现是需要每一个命令对通配符都理解的。

比如说剁死系统里运行“copy *.jpg aaa”。剁死这个懒得要死的家伙肯定啥也不管，只把 copy 叫醒，然后告诉他：“用户让我告诉你：‘*.jpg aaa’快干活去吧！然后 copy 就要去理解，*.jpg 就是所有的 .jpg 为扩展名的文件，所以用户给他的任务就是拷贝（因为他只会拷贝，别的任务不会叫他来）当前目录下的所 .jpg 文件和 JPG 文件和 Jpg 文件和 jPg 文件和 jpG 文件和……（谁让剁死不区分大小写呢。）拷贝到当前目录下的 aaa 目录去。这样做有什么不好呢？不好就是没个剁死下的命令，要想支持通配符，都要自己处理，浪费资源啊。

我们 linux 下的 shell 们就不这么懒了。

就拿 bash 来说吧，也一样支持通配符，同样也是“*”代表任意个任意字符，“?”代表某一个任意字符。不过，通配符的解释都是由 bash 来做的。

比如拷贝文件，运行“cp *.jpg ./aaa ”那么，这条命令输入进去之后，先交到 bash 这里，bash 一看有通配符那就需要自己进行解释，“*”代表任意长个字符嘛，那么这条命令的本意就是“cp 11111.jpg 22222.jpg 33333.jpg 44444.jpg ./aaa ”，也就是说，bash 用当前目录下的所有 jpg 文件的文件名代替了 *.jpg 这个字段，然后再叫醒 cp，把扩展后的参数传给他，cp 看到

的就是没有星号的命令了，翻译过来就是“复制当前目录下的 11111.jpg 和 22222.jpg 和 33333.jpg 和 44444.jpg 到当前目录下的 aaa 目录下。”这样做的好处是明显的：bash 下的各种软件都不许要自己处理通配符的问题，减少了不必要的重复开发。

当然，也有的时候是不需要 bash 对特殊符号进行扩展的。比如主人可能就是有个文件叫做“*.jpg”（没错！就叫这么个特殊的名字！在我们 linux 里你爱叫啥就啥，绝对不会跳出来个警告框说你不能起这种名称之类的，搞的主人好怕怕），要想复制着一个文件而不是所有的 jpg 文件那怎么办呢？也好办，就这样：“cp *.jpg ./aaa”在“*”前面加上一个“\”就可以了，“\”的意思就是后面的字符是个普通字符，告诉 bash 不要进行处理。

故事外的事——iwconfig

这节说到 iwconfig，那么就为懒得查 man 的同学们说说 iwconfig 的用法吧。

iwconfig 小程序是用来管理无线网络的，现在 wifi 的使用是越来越广泛了，很多人家里都使用无线路由器来共享网络链接，我家主人这里也是。在图行界面下，那个 nm-applet 工作的还是不错的，只是主人不想特意为了链接网络而去一下图形界面，所以就学习起 iwconfig 来。iwconfig 的主要任务就是负责把电脑接到某个无线接入点上。经常和 iwconfig 同时来工作的还有一个家伙，叫做 iwlist，这里是用来查找无线信号的。一般，都是先运行 iwlist scan，来扫描附近可以找到的无限信号，然后再选择其中的某一个接入。iwlist 得到的结果里最主要的是那个你意中的无线接入点的 essid，有了这个 id 才好跟 iwconfig 说你要接谁，如果不说的话，你只运行 iwconfig，那么他只是帮你列出现在机器上的无线网卡的状态——多半是单身状态，也就是说没有连接到任何无线接入点中。那么要想连接怎么办呢？只要知道了 essid，就运行 iwconfig wlan0 essid “xxxxxx”就可以了，就像你知道了对方姑娘叫啥，才好找人去她家说媒是不。这其中 wlan0 是你要是用的无线网卡（因为你可能有多个无线网卡，就像你有俩儿子，你得决定一下给那个儿子说媒嘛），xxxx 是那个你要链接的无线接入点的名字，也就是 essid。这样运行了之后，iwconfig 就会去试着让你的无线网卡跟对方眉来眼去一翻，然后鸿雁传书，互诉衷肠，心驰神往，最后喜结连理——你就能上网了。不过这是对方没什么要求的情况，有的时候对方会要点彩礼——一把钥匙。别急，不是房子的钥匙，不过自行车钥匙也不行，要的是能够解开对方

甜言蜜语的钥匙——解密用的 Key。毕竟无线这东西看不见摸不着，你俩聊的卿卿我我的搞不好就隔墙有耳，所以设个密码还是需要的。密码基本有两类，wep 的和 wpa 的。wpa 又有很多种，我们暂时就不具体讨论了，因为 iwconfig 处理不了那些，如果是 wep 的话，那就好办。链接的时候就运行 iwconfig wlan0 essid “xxxxxx” key yyyy-yyyy-yy 这样的格式就可以了。那些 yyyy 就是密码，就像俩人的信物一样。密码必须是 5 个字节的 16 位数字，具体是什么，那只有你知道咯。当然，除此之外 iwconfig 还有很多功能，比如设置网卡工作频率阿，设置网卡自己作为无线接入点啦之类的，这些东西，你还是去问那个纯爷们吧。如果你也不大愿意和那家伙交流，你也可以问那个命令，就是运行 iwconfig --help，让他自己告诉你应该怎么用。不过，虽然绝大多数命令都可以用 --help 来查看使用方法，但毕竟不是全部的命令都能这样，要最详细的说明，还是得靠 Man。

6.3 多彩的 Shell

主人逐渐的开始适应了命令行的操作，于是他有个想法，在这个黑漆漆的界面中，搭建起一个可用的环境，这样以后开机就可以不进图形界面，直接用命令行的软件来做各种事情，可以更高效，更快速。

他决定，用 7 天的时间来完成这件事情。

起初，主人来到混沌漆黑的命令行。界面是黑漆漆一片，ls 还会出些菱形的方块块，主人的手指游弋在键盘上面主人说：“要有中文！”就有了中文。

主人看中文是顺眼的，有中文，有英文，没有了乱码。这是头一日主人说：“要有声音！打破寂静的黑夜。”

主人就让命令行里发出美妙的乐声，这是第二日主人说：“要有窗，看到外面的世界。”于是有了浏览器，主人可以重新看到那些网络上的奇花异草，光怪陆离。

有了通讯工具，主人可以把这些光怪陆离，异草奇花分享给朋友们，这是第三日主人说：“要有色彩，有图，才有真相！”于是就有了图。

图片为黑白的世界带来了色彩，美好的，丑陋的，思念的，怀旧的，唯美的，憧憬的，现实的，抽象的，红色的，绿色的，蓝色的，黄色的，各种的图片，主人微笑了。这是第四日。

主人说：“要动起来，要鲜活的世界。”于是，就动起来了，这是第五日。

第六，七日，主人说，我累了，要休息，于是，就没有于是了，丫没开机。

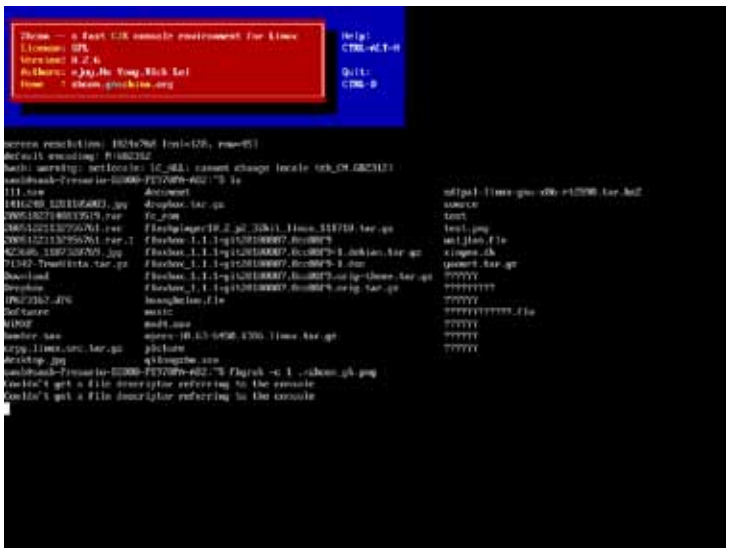
众人说：和着您主人就是传说中的上帝？？

我说：不是，上帝第七天才休息，主人第六天就休息了。

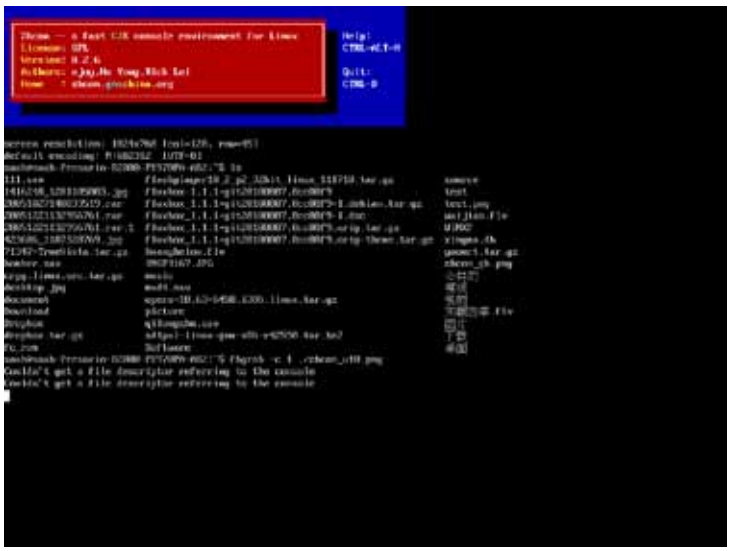
众人说：废话，那是上帝没赶上实行双休日。

咳咳，我估计这么说您还是听不懂，那我慢慢给您讲解。

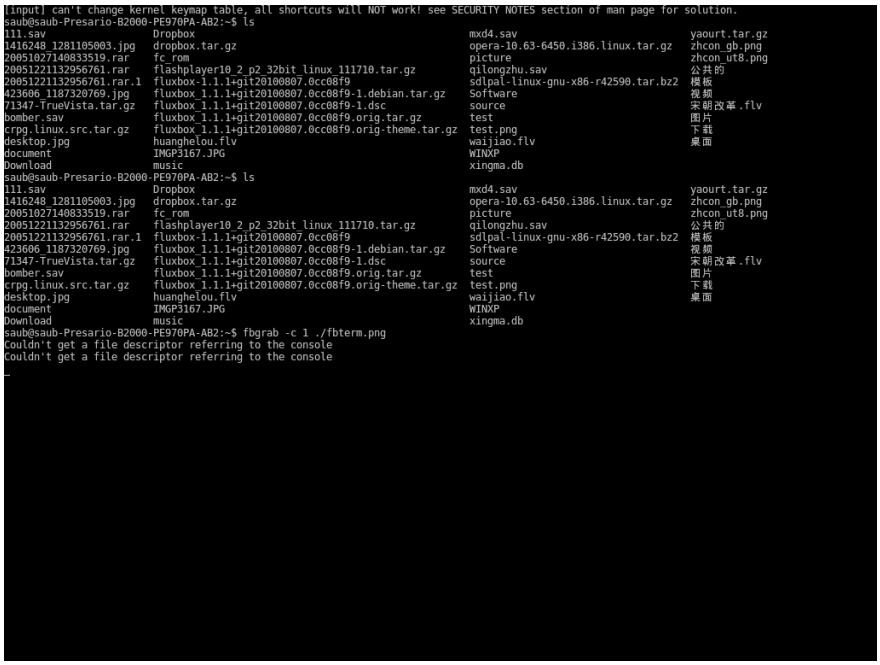
这周一，主人开始打造自己的命令行世界。各种命令对主人来说都已经很熟悉了，就算有些不大熟练的命令，也可以很轻松的找到纯爷们来讲解。但是一来到命令行里面遇到的第一个问题就是，没有中文的支持。运行下 ls，主目录下的图片啊，文档啊这些目录显示的都是一些菱形的方块。这也难怪，在伪终端下，中文环境由图形部门负责，可到了真正的终端界面中，这事情就没人管了。毕竟多数人还是要使用图形界面的嘛，像我主人这么神经质的人不多，所以我们没有更多的考虑字符界面的使用感受。不过主人现在需要字符下的中文环境，那我们也有人能够胜任，而且，还不只一个。



首先就是老牌的 zhcon，这个家伙有些岁数了，很久以前他就担当起让命令行下显示和输入中文的重任。他是一个外挂的中文环境，就像很久以前的“剁死”系统里的那个 UC-DOS 一样。要安装他很简单，找超级牛力就行。装上之后，在终端里（不能是图形界面下的伪终端哦。）输入 zhcon，就运行起来了。不过主人就这么运行了之后发现，这目录下的中文还是有问题。原来中文的地方都是菱形方块，现在倒好，改成一堆问号了。难道这 zhcon 的能力就是把方块改成问号？



想想也不是阿，于是主人运行了“zhcon --help”，问问 zhcon 这到底是怎么回事。zhcon 解释说，你要设置一下编码，我默认是使用 GB2312 的，你系统里的文件都是 UTF-8 的，那你运行的时候就要这样：“zhcon --utf8”才行。主人这才恍然大悟，心说你倒是早说啊。赶紧照着他说了运行，然后 ls 一下，果然，终于看见中文了。按下 ctrl-空格，哈哈，出现了输入法，虽然全拼不是那么好用，不过，命令行下，也凑和了。



zhcon 也是有不少后遗症的，启动了 zhcon 后，凡是一些需要有排版有格式的显示，比如 w3m 显示的网页啦，moc 显示的 mp3 播放介面啦，都会或多或少的乱掉。所以主人又找了更好的解决方案——fbterm。

fbterm 是一个基于 framebuffer 的终端。framebuffer 是一种字符界面下程序访问显卡的接口。他是能让命令行更加丰富多彩的关键的一环。通过 framebuffer，程序们可以直接向显卡的显存里面更新数据，也就是直接影响屏幕上显示的点和颜色，这就让在终端下显示图片成为了可能。不过由于程序是直接控制显卡的显存，并不经过显卡运算，所以显卡的运算能力是不能发挥的，所有的图形都是靠 CPU 运算好之后直接送到显卡显存里，所以 framebuffer 是比较消耗 CPU 的。framebuffer 以后我们还会经常看到，现在先说 fbterm。

fbterm 其实也可以算是个虚拟终端了，就跟图形界面下的那个 Gnome 终端一样，说不定还是一个学校毕业的呢。只不过 fbterm 是在终端下，用 framebuffer 来“画”出一个虚拟终端来，而 Gnome 终端是靠图形界面组的那些人来画虚拟终端。fbterm 的好处就是不像 zhcon 那样会搞乱屏幕的排版，所以主人自从用了 fbterm 之后，就再也没运行过 zhcon。不过 fbterm 只是一个能够显示中文的终端而已，他不是一个中文环境，所以，只能看到中文，输入是不行的。不过没关系，专门在命令行下的输入法，也是有的，和 fbterm 配合起来，就天衣无缝了。

关于中文输入法，首先就是来自中国的 yong，学名叫做小小输入法，这个我们的源里可没有，得去他的网站上下载并安装。装好之后，只要在运行 fbterm 的时候加上参数，指定输入法，就可以在 fbterm 中使用了，指定输入法用 -i 参数，就这样：“fbterm -i yong”。除了 yong，还有 fbterm_ucimf 可以使用，不过相对有点简陋。另外还可以使用 ibus-fbterm，看名字就知道了，是基于 ibus 的，这个相对好用些。

最终主人是选择了 fbterm，这不单是因为看上去顺眼点，更关键的是后来的境遇。

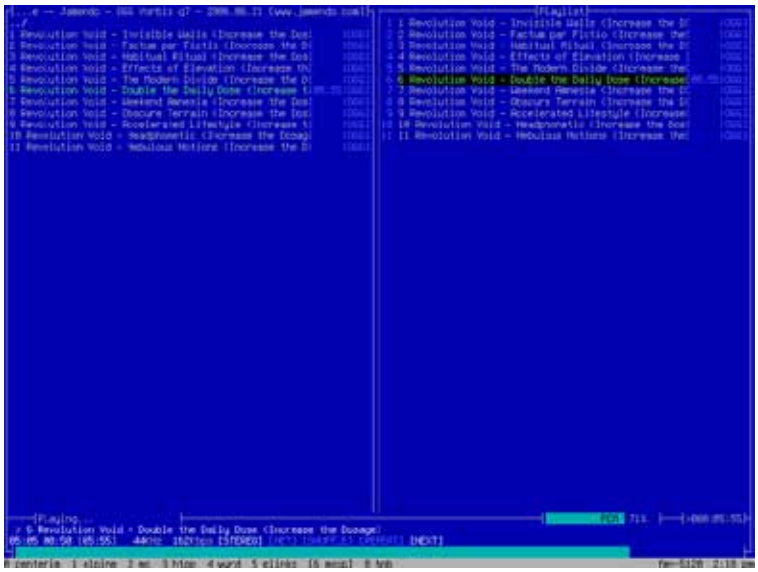
周二，主人得到了比较顺手的有中文环境的命令行之后，又开始踏上了新的寻觅旅程——寻觅音乐。

在图行界面下的时候，有很多的播放软件可以让主人浸泡在音乐的海洋里。那在字符界

面呢？想想也不应该有问题呀，放音乐又不需要图形界面是不是？字符界面下播放个音乐还是不成问题的，mplayer 就可以播放各种音频，还有个 mpg123 也可以放出声来。你知道 gnome 界面下，鼠标悬停在音乐文件上就可以播放出声音么？不知道？你 OUT 了。那个悬停播放的功能就是靠调用 mpg123 实现的。不过这俩虽然能马马虎虎弄出点动静，不过毕竟不专业，怎么也得有个播放列表，歌曲管理的功能吧？于是，主人选择了 moc。

moc 是个字符界面的音乐播放软件。有好奇而 OUT 的同学可能猜想了：字符界面播放音乐，那是不是要输入命令才播放音乐呢？比如输入 play，就播放，输入 stop 就暂停，输入 load xxxx.list 就导入播放列表，输入…… 反正干啥都得输入吧？这个，又 OUT 了吧？现在这么简介高效的年代，怎么可能又这么难用的东西呢？虽说是字符界面，但是不代表就不能有窗口！是的，你没听错，在字符界面下也能画出窗口来！！怎么画？用字符拼呗！！！

我们知道，ASCII 码中有一些特殊的字符，什么横杠阿，竖杠阿，拐弯杠阿……什么的。用这些特殊的符号，加上可控制的字符底色，可以拼接出又窗口效果的终端显示的界面。当然，如果每个程序都自己写代码去拼去肯定是很费事的，所以崇尚共享的我们 linux 系统提供了一套专门在字符界面下画窗口的函数库，叫做 ncurses，调用这个库画窗口就跟用 gtk+ 提供的接口在图形界面下画窗口类似，所以编程的人员不必在意怎么用各种字符拼出好看的窗口，只要调用 ncurses 就行了。ncurses 的前身是 curses，大名鼎鼎的 vi 就是用它实现的界面。moc 是一个基于 ncurses 的，字符界面的音乐播放软件，直接找超级牛力就可以安装，主人装了之后赶紧运行一下试试。虽然包名叫 moc，不过运行的命令是 mocp，运行起来之后就是这个样子：



左边是文件列表窗口，可以通过上下箭头选择相应目录，按回车进入，最上面的那个“..”表示上一级目录，这个都知道吧？不知道的面壁去。选到要听的 mp3 文件按 A 键将文件加入右边的播放列表。把 mp3 都加进来之后，用 TAB 键，切换左边的文件列表和右边的播放列表。按回车就可以播放了。然后还有些快捷键比较常用，左右键可以调节播放的进度，“,”“.”用来控制音量，s 键停止，b，n 跳到上一首 / 下一首歌，R 和 X 键用来控制重复播放和循环播放。（注意大小写的区别哦，所谓按 R，就是按下 shift 然后按下 r 键，也就是在终端打出大写的 R 的操作顺序。）

有人说，这播放软件把整个屏幕都占了，我还怎么干别的呀？简单，你有三个选择：

1. 不干别的了；

2. ctrl-alt-f2，换个终端；

当然，这两个方法都是比较笨笨的，呵呵。其实 mocp 是一个 server 和前端分离的软件，你可以按 q 键退出前端界面，然后该干啥干啥，音乐继续播放，想再回来就再运行 mocp。如果想彻底退出就按 Q。

好了，主人已经添加好播放列表，按下回车键，音箱里传来了悠扬的乐曲：“……老爷子 89 了，满面红光，别看就一个牙了，吃东西胃口很好，吃着吃着还塞牙了。”“就一个牙了怎么还塞牙？”“吃藕，套眼里了”……

周三这天，主人要解决一个关键的应用——上网。平时在图形界面下，最忙录的就是狐狸妹妹，基本上只要电脑开着，内存里就少不了狐狸妹妹的身影。这要是命令行下不能上网，那还不得憋屈死。不过在命令行毕竟是命令行，功能还是有限的，所以对浏览器的要求也不能太高是不是，简单的文字的网页还是没问题的，图片呢，实时的显示也不行，不过勉强可以调用别的程序来看一下，但是 flash 啦，在线的视频这类的就不用想了。好在主人比较通情达理，也就是想看看文字的东西和简单的一些图片，所以，有那么几个候选软件是可以满足主人要求的：w3m，lynx，links。

这三个都是可以直接叫超级牛力去装的，很省事。但经过主人的试用，lynx 和 links 对中文的支持都相对差点，并且他们都不能显示图片，所以被 pass 了，w3m 华丽胜出。不过 w3m 也不是天生就会显示图片，他需要一些工具，这些工具被打成一个包放在软件源里，包的名字叫做 w3m-img，主人让超级牛力给 w3m 搬来了这个工具包之后，w3m 才拥有了显示网页上图片的能力。

当然，w3m 显示图片跟 moc 显示窗口可不一样，可不是拿特殊字符拼出图片，要那样的话，整个屏幕，顶多拼个超级玛丽出来，连蘑菇都没地方显示了。要显示正经的 jpg 这样的图片，还是需要 framebuffer 的支持（当然，一些不正经的图片也一样需要 framebuffer）。基本上 framebuffer 是能够使命令行界面变得多姿多彩的最基本最重要的条件。主人要访问的网页无疑基本都是中文的，所以需要能够支持中文的终端，咱之前说过了，就是 fbterm。于是，主人就在 fbterm 下运行起了 w3m。

不过，天有不测风云日，人有无意失手时，软件也有偶尔不大正常的时候。虽然在某些系统某些机器上，fbterm 下的 w3m（前提是装了 w3m-img 哦）确实可以正常的显示出图片，不过不知道为什么在我们这里，这个 w3m 和 fbterm 配合了半天也没配合上，他们一运行起来就听着他们俩不停的吵吵：“fbterm，快点把 fb0 设备给我，我要显示网页上这个图片”“你怎么又要啊，我这边的中文还没显示好呢。”“你怎么那么多中文要显示啊？？”“你这不废话么，我显示的都是你的网页，显示多少中文还不是你所了算。”“别急别急，你显示完了赶紧的给我用。”“等等，等等，再有 3 毫秒就完了”“我倒，主人翻篇了。刚才那个图片翻过去了，白算了半天。”“好了我用完了，给你用去吧”“用什么用，他已经翻到下一页了，又一个新的图片，我先得解码看看这个图片里是什么才知道该显示什么嘛。”……

总之，这俩从来没对上过，于是主人就没看到图片。不过后来主人还是找到了解决办法，那就是，换了另一个终端，其实跟 fbterm 还有点关系，叫做 jfbterm，也一样可以支持中文。在 jfbterm 下，w3m 终于可以正常显示图片了。

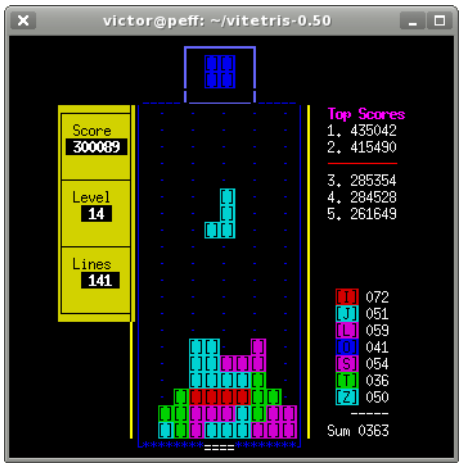
周四，主人又一次摄影归来，插上 SD 卡，运行 mount 命令挂在并且把照片 cp 到主目录中的照片文件夹。之后，就开始研究命令行下到底用什么来看照片了。要看照片无疑还是需要 framebuffer 的支持，这个咱就不说了。那么那个能够利用 framebuffer 设备来显示图片的软件是谁呢？那可是一个大名鼎鼎的家伙——FBI ！！

别激动，这个 FBI 不是某大叔的啥啥调查局，只是 linux 下的一个 Fram Buffer Image viewer，也就是基于 framebuffer 的图像查看器而已。这个查看器可以查看各种常见的图片格式，全屏显示，缩放，幻灯片模式播放，旋转，都没问题，当然更复杂的什么调节对比度、亮度啥的自然就不行了，毕竟人家只是个查看器嘛。主人拷贝好了照片后，用 cd 命令切换进入存着照片的目录，运行了“fbi ./*.JPG”，这个命令大家应该能看懂吧，就是让 fbi 去查看当前目录下的，所有的 JPG 文件的意思。顺便复习一下，* 这个通配符是由 bash 来处理的，就是说，主人输入了这个命令后，bash 先处理一下，把命令改成“fbi ./123.JPG 456.JPG GSMLY.JPG WTL.JPG CJK.JPG BDYJY.JPG FDA.JPG”等等等等，然后再按照这个命令去叫醒 fbi，告诉他要去显示这么多图片。然后 fbi 就起床，一个一个的显示这些图片，当主人按 -、+ 键的时候就对图片进行缩小，放大。按下 PageDown 键就播放下一张，偶尔遇到竖着拍的照片可能还需要按 R 或者 L 旋转一下，总之，主人在悠然轻松的欣赏照片的过程中度过了一晚上。

光看静态的照片还是有些不过瘾的，于是周五，主人拷来了一个叫 RPT.rmvb 的文件，想要看片了。那命令行下能看片么？答案依然是肯定的，当然，还是离不开 framebuffer。平时在图形界面下主人经常用来看片的播放器是 SMplayer，当初咱们介绍这个家伙的时候就说过，他只是个前端界面，真正在后面默默无闻的进行播放工作的是 Mpayer，Smplayer 能播放什么文件完全取决于 mplayer 能播放什么文件，只是因为 mplayer 总是在后台，人们都忽略了它的存在。现在，到了字符界面，mplayer 终于可以从后台来到前台了。主人直接运行了 mplayer RPT.rmvb ，于是，一个动态的画面出现在了黑漆漆的屏幕上，一个醒目的字幕写着：肉片汤的做法 ……

除了硬盘里现成的视频文件以外，mplayer 还可以播放 mms 流媒体，也就是一些在线的

视频也是可以播放的。主人找到了一些网上的电视台的流媒体地址，什么 CCAV5 阿，CCAV8 阿，这 AV 那 AV 的，反正我是不知道什么内容，但是主人看的很起劲。好像他屋里也没有那个叫做 TV 的东西，于是就拿电脑当作 TV 了。不过每次总是输入很长的 mms 地址很麻烦，所以主人就写了个脚本，把脚本命名为 tv.sh，运行的时候后面加上 CCAV5 阿 CCAV8 阿作为参数，想看电视的时候就晕行 tv.sh CCAV5，然后就可以坐在电脑前享受着没有 xorg 运行的 linux 系统带来的电视节目了。



到了周六，主人休息了，周日也没来找我们。不过后来的后来还是有故事发生的，主人竟然开始在字符界面下玩游戏了。不过为了同学们好好学习，这里就不多做介绍了，就给两个线索吧，有兴趣的同学可以研究 vitetris。

```
apt-get install bsdgame
```